學號：B06901180　系級：電機三　　　姓名：鄭謹譯

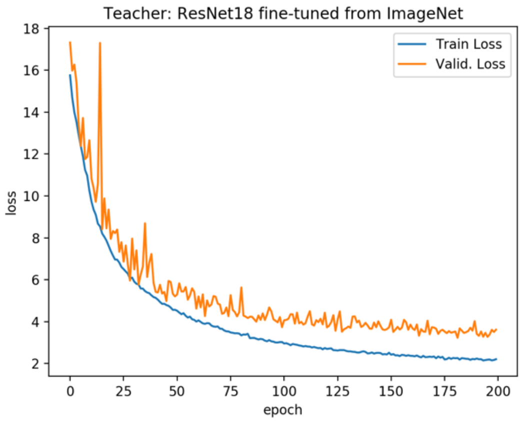
1. 請從 Network Pruning/Quantization/Knowledge Distillation/Low Rank Approximation 選擇兩個方法(並詳述)，將同一個大 model 壓縮至同等數量級，並討論其 accuracy 的變化。 (2%)　Cooperate with 筠婕 （ Same Archiecture ）

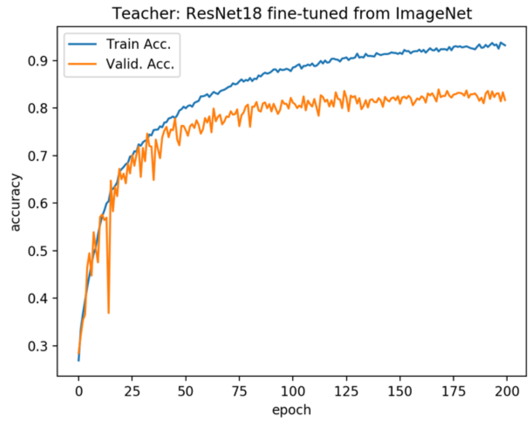
   Architecture Design :

```
StudentNet(
  (cnn): Sequential(
    (0): Sequential(
      (0): Conv2d(3, 16, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6()
      (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    )
    (1): Sequential(
      (0): Conv2d(16, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6()
      (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    )
    (2): Sequential(
      (0): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6()
      (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    )
    (3): Sequential(
      (0): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6()
      (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    )
    (4): Sequential(
      (0): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6()
      (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    )
    (5): Sequential(
      (0): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6()
      (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    )
    (6): Sequential(
      (0): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6()
      (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    )
    (7): Sequential(
      (0): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6()
    )
    (8): AdaptiveAvgPool2d(output_size=(1, 1))
  )
  (fc): Sequential(
    (0): Linear(in_features=256, out_features=256, bias=True)
    (1): Linear(in_features=256, out_features=11, bias=True)
  )
)
Size of parameters =  2233995
```

   Knowledge Distillation : Student net same as AD, Teacher net : (ResNet18) ImageNet pretrained & fine-tune

   Accuracy on Validation Set : 0.83906





Quantization : 8-bit quantization from (ResNet18) ImageNet pretrained & fine-tune

```
Loading test data...
Start testing
Correct:  3027  Total:  3430  Accuracy:  0.8825072886297376
```

Accuracy on Validation Set : 0.88251

```
(base) chinyi0523@SpeechLab531:~/hw7-chinyi0523$ bash size.sh
Size of Model: in bytes
44788712        teacher_resnet18.bin
08956370        Report_KD.bin
11208784        Report_QW.bin
```

大Model : teacher_resnet18.bin

由結果發現Quantization的結果較佳，推測因為Quantization只有壓縮儲存方式，整體Model仍然是teacher_resnet18.bin，只有少許失真（助教給出的teacher_nresnet18.bin acc為0.8841），準確率沒有特別大的變化。Knowledge Distillation的student model參數量為2,233,995，為QW和Teacher 11,182,155的1/5，故有些資訊必然流失，準確率也下降到 0.83906。

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. 架構為ResNet18

y. Student net architecture and # of parameters: 256779 parameters

```
Loading validation data...
StudentNet(
  (cnn): Sequential(
    (0): Sequential(
      (0): Conv2d(3, 16, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6()
      (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    )
    (1): Sequential(
      (0): Conv2d(16, 16, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=16)
      (1): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6()
      (3): Conv2d(16, 32, kernel_size=(1, 1), stride=(1, 1))
      (4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    )
    (2): Sequential(
      (0): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=32)
      (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6()
      (3): Conv2d(32, 64, kernel_size=(1, 1), stride=(1, 1))
      (4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    )
    (3): Sequential(
      (0): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=64)
      (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6()
      (3): Conv2d(64, 128, kernel_size=(1, 1), stride=(1, 1))
      (4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    )
    (4): Sequential(
      (0): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=128)
      (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6()
      (3): Conv2d(128, 256, kernel_size=(1, 1), stride=(1, 1))
    )
    (5): Sequential(
      (0): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=256)
      (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6()
      (3): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1))
    )
    (6): Sequential(
      (0): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=256)
      (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6()
      (3): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1))
    )
    (7): Sequential(
      (0): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), groups=256)
      (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (2): ReLU6()
      (3): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1))
    )
    (8): AdaptiveAvgPool2d(output_size=(1, 1))
  )
  (fc): Sequential(
    (0): Linear(in_features=256, out_features=11, bias=True)
  )
)
Size of parameters =  256779
```

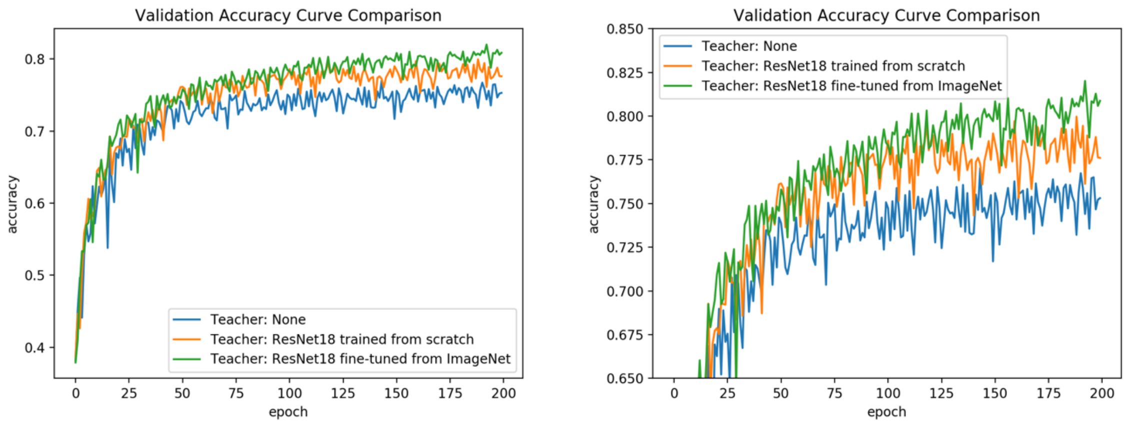
a. Teacher net (ResNet18) from scratch: 80.09%

b. Teacher net (ResNet18) ImageNet pretrained & fine-tune: 88.41%

c. Your student net from scratch: 77.02%

d. Your student net KD from (a.): 79.97%

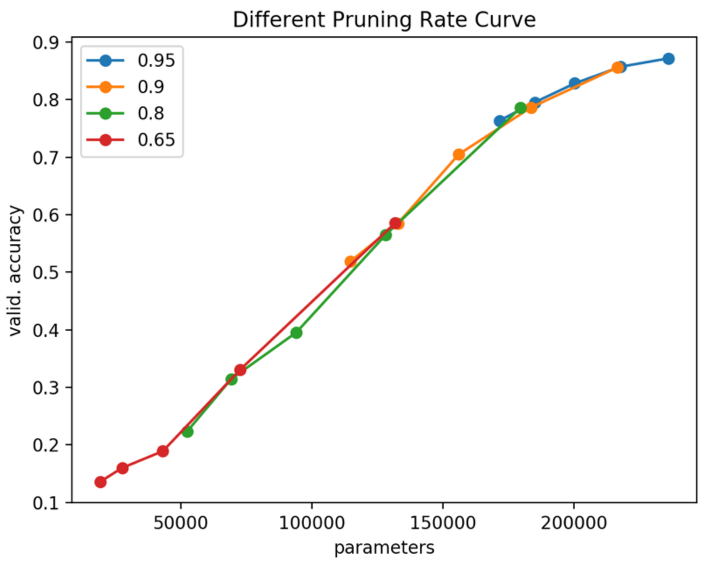
e. Your student net KD from (b.): 82.01%



上左圖為Validation Accuracy Training Curve，上右圖為左圖放大，c準確率（藍線）在50 epoch 以後準確率就與d,e有明顯落差，最後結果也較低。在125epoch後漸漸可以看出e與d的差距，最後結果e>d>c。

基本上仍可看出Fine tuned的結果較佳，因為teacher net可以提供soft label的資訊，能看出不同label之間的相關性，相較於train from scratch 只有hard label，teacher net給student更多學習的資料，因此c的結果較d,e差。同理可應證於a,b上。而d,e的差別應該來自於a,b本來的準確率就有8%差距，d學的比e差也在預測中。

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%)**



四種pruning rate 0.95,0.9,0.8,0.65

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)