Machine Learning 2020 - HW7 Report

學號: b07902064 系級: 資工二 姓名: 蔡銘軒

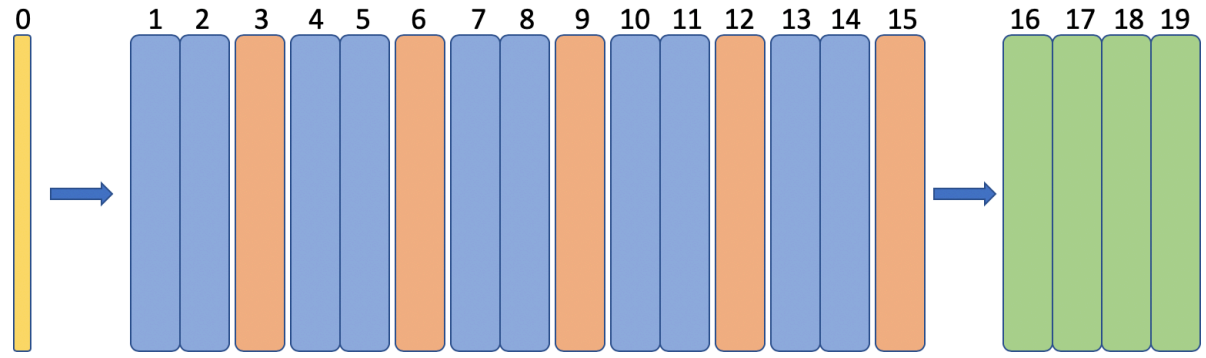
1. 請從 Network Pruning/Quantization/Knowledge Distillation/Low Rank Approximation 選擇兩個方法(並詳述)，將同一個大 model 壓縮至同等數量級，並討論其 accuracy 的變化。 (2%) 以下三題只需要選擇兩者即可，分數取最高的兩個。

   大model：我在HW3的best model

   validation set accuracy: 88.51%

   model state_dict大小: 177530883 bytes

   架構：



   0: input (192x192x3) / 1: Conv2d (kernel 3x3, channel 64) / 2: Conv2d (kernel 3x3, channel 128)

   3: Maxpool (2x2) / 4: Conv2d (kernel 3x3, channel 256) / 5: Conv2d (kernel 3x3, channel 256)

   6: Maxpool (2x2) / 7: Conv2d (kernel 3x3, channel 256) / 8: Conv2d (kernel 3x3, channel 256)

   9: Maxpool (2x2) / 10: Conv2d (kernel 3x3, channel 512) / 11: Conv2d(kernel 3x3, channel 512)

   12: Maxpool (2x2) / 13: Conv2d (kernel 3x3, channel 512) / 14: Conv2d(kernel 3x3, channel 512)

   15: Maxpool (2x2) / 16: Linear (512x6x6, 2048) / 17: Linear (2048, 2048)

   18: Linear (2048, 1024) / 19: Linear (1024, 11)

   Low Rank Approximation(Architecture Design):

   validation set accuracy: 79.91% (train 250個 epoch 並記錄最佳)

   model state_dict大小: 1119767 bytes

   架構 (from torchsummary):

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 32, 256, 256]	896
BatchNorm2d-2	[-1, 32, 256, 256]	64
ReLU-3	[-1, 32, 256, 256]	0
MaxPool2d-4	[-1, 32, 128, 128]	0
Conv2d-5	[-1, 32, 128, 128]	320
BatchNorm2d-6	[-1, 32, 128, 128]	64
ReLU-7	[-1, 32, 128, 128]	0
Conv2d-8	[-1, 64, 128, 128]	2,112
MaxPool2d-9	[-1, 64, 64, 64]	0
Conv2d-10	[-1, 64, 64, 64]	640
BatchNorm2d-11	[-1, 64, 64, 64]	128
ReLU-12	[-1, 64, 64, 64]	0
Conv2d-13	[-1, 128, 64, 64]	8,320
MaxPool2d-14	[-1, 128, 32, 32]	0
Conv2d-15	[-1, 128, 32, 32]	1,280
BatchNorm2d-16	[-1, 128, 32, 32]	256
ReLU-17	[-1, 128, 32, 32]	0
Conv2d-18	[-1, 128, 32, 32]	16,512
MaxPool2d-19	[-1, 128, 16, 16]	0
Conv2d-20	[-1, 128, 16, 16]	1,280
BatchNorm2d-21	[-1, 128, 16, 16]	256
ReLU-22	[-1, 128, 16, 16]	0
Conv2d-23	[-1, 256, 16, 16]	33,024
Conv2d-24	[-1, 256, 16, 16]	2,560
BatchNorm2d-25	[-1, 256, 16, 16]	512
ReLU-26	[-1, 256, 16, 16]	0
Conv2d-27	[-1, 256, 16, 16]	65,792
Conv2d-28	[-1, 256, 16, 16]	2,560
BatchNorm2d-29	[-1, 256, 16, 16]	512
ReLU-30	[-1, 256, 16, 16]	0
Conv2d-31	[-1, 256, 16, 16]	65,792
Conv2d-32	[-1, 256, 16, 16]	2,560
BatchNorm2d-33	[-1, 256, 16, 16]	512
ReLU-34	[-1, 256, 16, 16]	0
Conv2d-35	[-1, 256, 16, 16]	65,792
AdaptiveAvgPool2d-36	[-1, 256, 1, 1]	0
Linear-37	[-1, 64]	16,448
Linear-38	[-1, 11]	715

為了壓縮model的大小，我簡化了大model的架構。例如大model的channel數量從64開始增加至512，但我發現在小model上使用512個channel，會因為最後要壓平傳進 FC layer 時造成 FC layer 增加很多參數，因此我改成從32個channel，增加到256個。而depth-wise與point-wise的部分則參考助教的sample code，在第一層沒有做拆解，後面的convolutional layer都有進行depth-wise與point-wise的處理，降低參數量。

FC 層原先試著在大 model 的 FC 層之間加入一層參數比較少的，實作教授上課影片內提到的 FC 層的 Low rank approximation，但正確率並沒有表現得很好。最後選擇只留兩層反而得到更好的表現，且參數也更少。

Knowledge Distillation:

model: 我使用與Low Rank相同的小model來進行測試。

validation set accuracy: 84.95% (train 250個 epoch 並記錄最佳)

訓練KD的方式與助教的 sample code 方式相同，在程式碼部分與前者大致相同，除了 Loss function 現在不只考慮小 model 本身預測與實際答案的 cross entropy，也考慮小 model 與大 model 預測結果的分布差異。我嘗試不同 alpha 值來改變兩者之間的比重，根據預測表現最後還是選擇 alpha = 0.5。這裡使用與上題相同 model 的想法是可以比較使用小 model 直接學習正確答案，以及學習大 model 的預測分佈，兩者之間表現的差異。從結果可以看到學習大 model 的行為比起直接學習正確答案表現進步不少。因為小 model 本身能力比較不足，要直接跟大 model 學習一樣的東西自然難以競爭。另一方面因為大 model 的輸出是比較連續的分佈，不是跟 true label 一樣是 one hot vector，我認為小 model 學習這樣的東西是比較容易的。因此在大 model 本身表現良好的情況下，小 model 也能有樣學樣得到不錯的結果。

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: 288,907

   使用的小 model 架構與上題相同，在此就不再附一次圖佔空間。

   以下關於小 model 的validation accuracy 都是 train 250 個 epoch 中紀錄最高者。

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

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

   c. Your student net from scratch: 79.91%

   d. Your student net KD from (a.): 84.34%

   e. Your student net KD from (b.): 81.08%

   大 model ResNet18 有比較良好的架構跟能力，反之小 model 本身參數就比較少，架構也沒有經過嚴謹的設計，表現自然有差 (88.41 % vs 79.91%)，但小 model 卻跟 ResNet18 from scratch 有相近的結果，我認為應該是 ResNet18 from scratch 沒有很 train 的很好或是還沒收斂。

   在 d. 的部分，小 model 除了有自己在 training set 上學習，也有跟著學習大 model 的預測分佈。我認為後者對於小 model 來說是比較容易學習的，因為他不是像答案一樣的one hot vector，而是比較連續的分佈。這樣小 model 在學習大 model 時，除了學到預測正確的 label，也有學到大 model 的"思考"，在大 model 本身表現不錯的情況下，小 model 的表現也提升不少 (提升5%左右)。

   在e.部分， 因為學習的大 model 本身就沒有很強，連小 model 直接學都能做到同等級的表現，因此加上KD只有稍微提昇表現。在這裡小 model 甚至表現得比大 model 略好。我認為是因為小 model 並非只跟大 model 學習，也有保留自己在 training set上的訓練結果，因此除非學習的大 model 表現太差，讓小 model 跟著學反而覺得看到許多雜訊，否則使用 KD 多少有輔助學習的效果，因此最後表現得比原本的大 model 也是可能的。

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%)

   略

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: 81.42%

   b. 將 CNN model 的 Convolution Layer 換成參數量接近的 Depthwise & Pointwise 後的 accuracy: 80.53%

   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): 79.71%

   以上的正確率都是 train 250 個 epoch 並記錄最佳者。

   model 說明:

   原始 CNN: 共有7層convolutional layer，channel 數量分別為 16, 32, 64, 64, 128, 128, 128，在第 1, 2, 4, 6, 7 層進行MaxPool。訓練參數量: 321,291，model 大小: 1.23 MB

   Depthwise & Pointwise CNN: 共有 8 層，channel 數量分別為 32, 64, 128, 128, 256, 256, 256, 256。除了第一層沒有做拆解，後面每一層都做了 depthwise & pointwise 的處理。在第1, 2, 4, 6, 8 層進行 MaxPool。訓練參數量: 274,571，model 大小: 1.05 MB

Group Convolution Layer: 共有 7 層，架構與原始 CNN 相同，修改的部分為從第二層開始，都使用 groups = 16 來減少參數量。訓練參數量: 292,971，model 大小: 1.12 MB

從這題的實驗可以發現使用 Depthwise & Pointwise 的處理可以大幅降低參數量。為了使 (b) 的 model 有與 (a) 相近的 model 大小與參數量，我把 channel 數量都提高兩倍且多加入了一層，但參數量還是不及 (a)。在 accuracy 的部分，發現正確率並沒有明顯下降，我認為可能跟增加 channel 數量以及整體層數

有關。我嘗試只將 (a) 的 model 做 Depthwise & Pointwise 分解，其餘架構維持相同，結果參數量大幅減少，正確率卻也降低比較多，無法突破 80%。

Accuracy的降低在 (c) 比較明顯。我認為這跟我在 (b) 做的實驗類似，維持 (a) 的架構，只對 convolution layer 做拆解，雖然這裡沒有拆到跟 (b) 一樣，但正確率也無法突破 80%。另外也可以觀察到 (c) 的拆解降低的參數量較少，拆解後 model 大小還是在 1 MB 左右，比較沒有空間讓我增加其他 layer 來增加 model 的能力。