Machine Learning - Homework 7

資工四 B05902023 李澤諺

May 21, 2020

在本次作業中，首先，我使用了助教所提供的 teacher net 和 student net 進行 knowledge distillation，其中，我使用了助教所提供的 data augmentation，並使用了 Adam 進行第一次的訓練，其中 batch size 爲 32，learning rate 爲 0.002，訓練了 150 個 epoch，接著，再使用 SGD 進行第二次的訓練，其中 batch size 爲 32，learning rate 爲 0.001，訓練了 50 個 epoch，以此得到最後的 model，其 size 與 accuracy 如下表所示：

Size	Train	Validation	Test	
			Public	Private
1MB	0.90351	0.82741	0.85236	0.84109

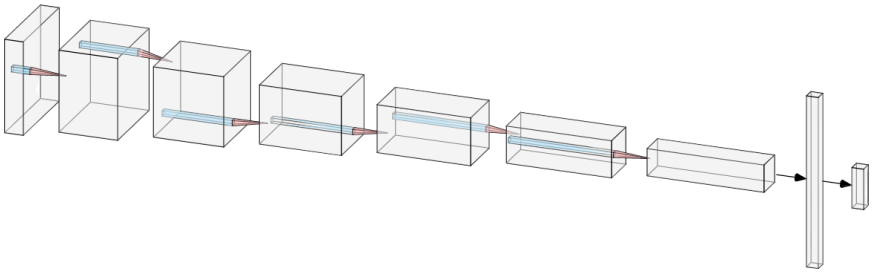
最後，我使用了助教所提供的 encode8/decode8 函式，進行 parameter quantization，所得到的 model size 與 accuracy 變化如下：

Size	Train	Validation	Test	
			Public	Private
263KB	0.96361	0.82653	0.85475	0.84289

(2、3、4 擇二寫即可，都寫的話取最高兩項加總。)

1. (2%) 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。

首先，我實作了一個簡單的 CNN (稱之爲 CNN 1)，其架構如下：

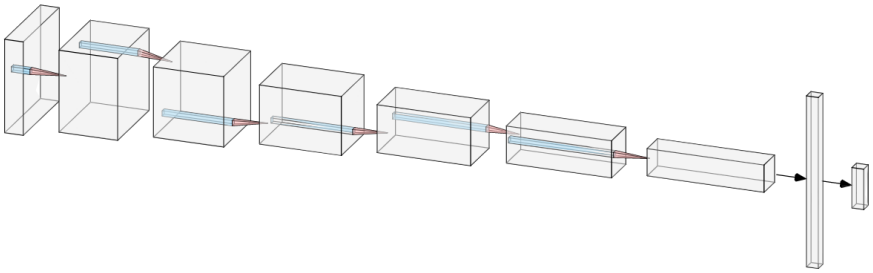


Conv2d(3 , 32 , kernel_size = 3 , stride = 1 , padding = 2)
BatchNorm2d(32)
ReLU()
MaxPool2d(2)
Conv2d(32 , 64 , kernel_size = 3 , stride = 1 , padding = 2)
BatchNorm2d(64)
ReLU()
MaxPool2d(2)
Conv2d(64 , 128 , kernel_size = 3 , stride = 1 , padding = 2)
BatchNorm2d(128)
ReLU()
MaxPool2d(2)
Conv2d(128 , 256 , kernel_size = 3 , stride = 1 , padding = 2)
BatchNorm2d(256)
ReLU()
MaxPool2d(2)
Conv2d(256 , 512 , kernel_size = 3 , stride = 1 , padding = 2)
BatchNorm2d(512)
ReLU()
MaxPool2d(2)
Conv2d(512 , 1024 , kernel_size = 3 , stride = 1 , padding = 2)
BatchNorm2d(1024)
ReLU()
MaxPool2d(2)
Linear(25600 , 11 , bias = True)

在訓練過程中，我使用了助教所提供的 data augmentation 進行訓練，並且，我使用了 Adam 進行第一次的訓練，其中 batch size 爲 32，learning rate 爲 0.002，訓練了 150 個 epoch，接著，再使用 SGD 進行第二次的訓練，其中 batch size 爲 32，learning rate 爲 0.001，訓練了 50 個 epoch，以此得到最後的 model，其 parameter 數量和 accuracy 如下表所示：

Parameter	Train	Validation	Test	
			Public	Private
6573835	0.95479	0.80496	0.83682	0.81959

接著，我將 CNN 1 中的 convolutional layer 替換爲 depthwise + pointwise convolutional layer (稱之爲 CNN 2)，其架構如下：



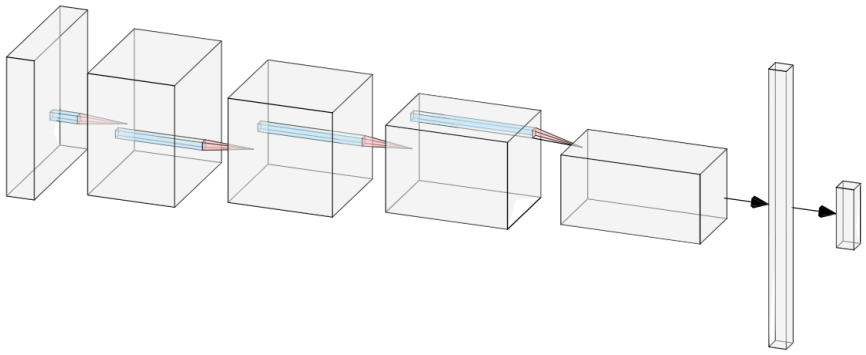
Conv2d(3 , 32 , kernel_size = 3 , stride = 1 , padding = 2)
BatchNorm2d(32)
ReLU()
MaxPool2d(2)
Conv2d(32 , 32 , kernel_size = 3 , stride = 1 , padding = 2 , groups = 32)
BatchNorm2d(32)
ReLU()
Conv2d(32 , 64 , kernel_size = 1)
MaxPool2d(2)
Conv2d(64 , 64 , kernel_size = 3 , stride = 1 , padding = 2 , groups = 64)
BatchNorm2d(64)
ReLU()
Conv2d(64 , 128 , kernel_size = 1)
MaxPool2d(2 , padding = 1)
Conv2d(128 , 128 , kernel_size = 3 , stride = 1 , padding = 2 , groups = 128)
BatchNorm2d(128)
ReLU()
Conv2d(128 , 256 , kernel_size = 1)
MaxPool2d(2 , padding = 1)
Conv2d(256 , 256 , kernel_size = 3 , stride = 1 , padding = 2 , groups = 256)
BatchNorm2d(256)
ReLU()
Conv2d(256 , 512 , kernel_size = 1)
MaxPool2d(2 , padding = 1)
Conv2d(512 , 512 , kernel_size = 3 , stride = 1 , padding = 2 , groups = 512)
BatchNorm2d(512)
ReLU()
Conv2d(512 , 1024 , kernel_size = 1)
MaxPool2d(2 , padding = 1)
Linear(50176 , 11 , bias = True)

在訓練過程中，我使用了助教所提供的 data augmentation 進行訓練，並且，我使用了 Adam 進行第一次的訓練，其中 batch size 爲 32，learning rate 爲 0.002，

訓練了 150 個 epoch，接著，再使用 SGD 進行第二次的訓練，其中 batch size 爲 32，learning rate 爲 0.001，訓練了 50 個 epoch，以此得到最後的 model，其 parameter 數量和 accuracy 如下表所示：

Parameter	Train	Validation	Test	
			Public	Private
1265163	0.84239	0.78280	0.81171	0.78554

最後，我使用訓練好的 CNN 1 作爲 teacher net，並實作另一個 layer 數目較少的 CNN (稱之爲 CNN 3) 作爲 student net，以進行 kownledge distillation，CNN 3 的架構如下：



Conv2d(3 , 32 , kernel_size = 3 , stride = 1 , padding = 1)
BatchNorm2d(32)
ReLU()
MaxPool2d(2)
Conv2d(32 , 64 , kernel_size = 3 , stride = 1 , padding = 1)
BatchNorm2d(64)
ReLU()
MaxPool2d(2)
Conv2d(64 , 128 , kernel_size = 3 , stride = 1 , padding = 1)
BatchNorm2d(128)
ReLU()
MaxPool2d(2)
Conv2d(128 , 256 , kernel_size = 3 , stride = 1 , padding = 1)
BatchNorm2d(256)
ReLU()
MaxPool2d(2)
Linear(65536 , 11 , bias = True)

在訓練過程中，我使用了助教所提供的 data augmentation 進行訓練，並且，我使用了 Adam 進行第一次的訓練，其中 batch size 爲 32，learning rate 爲 0.002，

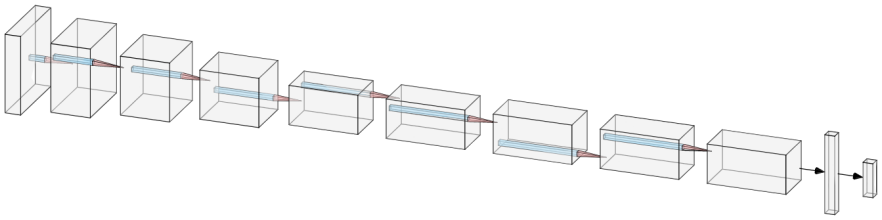
訓練了 150 個 epoch，接著，再使用 SGD 進行第二次的訓練，其中 batch size 爲 32，learning rate 爲 0.001，訓練了 50 個 epoch，以此得到最後的 model，其 parameter 數量和 accuracy 如下表所示：

Parameter	Train	Validation	Test	
			Public	Private
1110283	0.84543	0.78921	0.82307	0.80047

2. (2%) 請嘗試比較以下 accuracy (兩個 Teacher Net 由助教提供) 以及 Student Net 的總參數量以及架構，並嘗試解釋爲甚麼有這樣的結果。你的 Student Net 的參數量必須要小於 Teacher Net 的參數量。

x. Teacher net architecture and of parameters: torchvision ' s ResNet18, with 11182155 parameters.
y. Student net architecture and of parameters:
a. Teacher net (ResNet18) from scratch: 80.09%.
b. Teacher net (ResNet18) ImageNet pretrained fine-tune: 88.41%.
c. Your student net from scratch:
d. Your student net KD from (a.):
e. Your student net KD from (b.):

我使用了助教所提供的 student net，其架構如下：



Conv2d(3 , 16 , kernel_size = 3 , stride = 1 , padding = 1)
BatchNorm2d(16)
ReLU6()
MaxPool2d(2 , stride = 2 , padding = 0)
Conv2d(16 , 16 , kernel_size = 3 , stride = 1 , padding = 1 , groups = 16)
BatchNorm2d(16)
ReLU6()
Conv2d(16 , 32 , kernel_size = 1)
MaxPool2d(2 , stride = 2 , padding = 0)
Conv2d(32 , 32 , kernel_size = 3 , stride = 1 , padding = 1 , groups = 32)
BatchNorm2d(32)
ReLU6()
Conv2d(32 , 64 , kernel_size = 1)
MaxPool2d(2 , stride = 2 , padding = 0)
Conv2d(64 , 64 , kernel_size = 3 , stride = 1 , padding = 1 , groups = 64)

BatchNorm2d(64)
ReLU6()
Conv2d(64 , 128 , kernel_size = 1)
MaxPool2d(2 , stride = 2 , padding = 0)
Conv2d(128 , 128 , kernel_size = 3 , stride = 1 , padding = 1 , groups = 128)
BatchNorm2d(128)
ReLU6()
Conv2d(128 , 256 , kernel_size = 1)
Conv2d(256 , 256 , kernel_size = 3 , stride = 1 , padding = 1 , groups = 256)
BatchNorm2d(256)
ReLU6()
Conv2d(256 , 256 , kernel_size = 1)
Conv2d(256 , 256 , kernel_size = 3 , stride = 1 , padding = 1 , groups = 256)
BatchNorm2d(256)
ReLU6()
Conv2d(256 , 256 , kernel_size = 1)
Conv2d(256 , 256 , kernel_size = 3 , stride = 1 , padding = 1 , groups = 256)
BatchNorm2d(256)
ReLU6()
Conv2d(256 , 256 , kernel_size = 1)
AdaptiveAvgPool2d(1)
Linear(256 , 11)

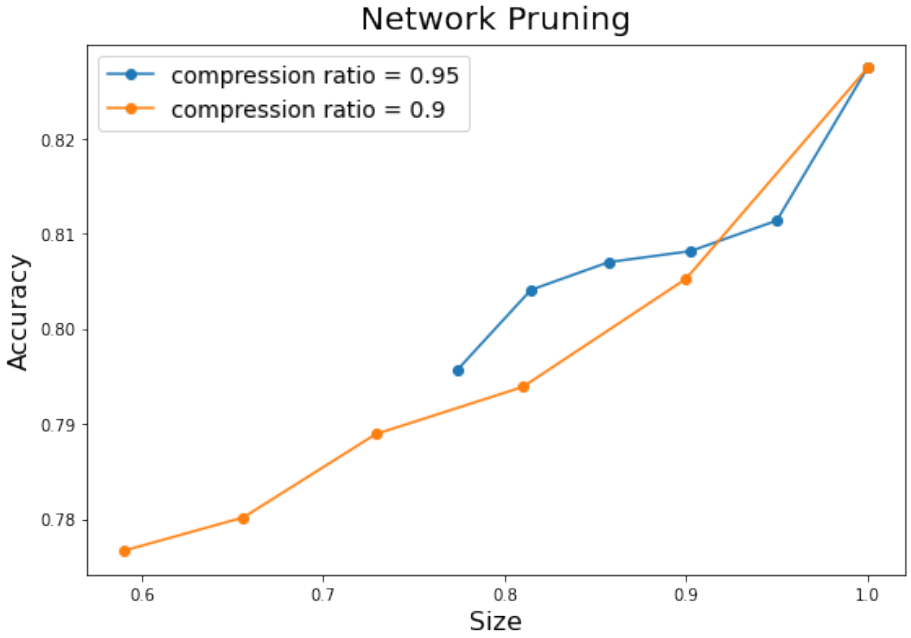
根據 torch-summary，以上的 student net 中總共有 256779 個 parameter。接著，我使用了 c、d、e 的方法訓練 student net，在這三個方法中，我皆使用了助教所提供的 data augmentation 進行訓練，並且，我使用了 Adam 進行第一次的訓練，其中 batch size 爲 32，learning rate 爲 0.002，訓練了 150 個 epoch，接著，再使用 SGD 進行第二次的訓練，其中 batch size 爲 32，learning rate 爲 0.001，訓練了 50 個 epoch，以此得到最後的 model，其 accuracy 如下表所示：

Method	Train	Validation	Test	
			Public	Private
c	0.92439	0.80000	0.81709	0.80107
d	0.88131	0.80816	0.83741	0.81123
e	0.90351	0.82741	0.85236	0.84109

由此可以看出，使用 knowledge distillation 所得到的 model 其 performance 比直接訓練而得的 model 還要高了一些，其原因大概是因爲使用 knowledge distillation 可以爲 model 提供更多的 information，進而協助 model 提高 performance，而在 kowledge distillation 之中，使用 pretrained 並 fine-tune 過的 ResNet-18 作爲 teacher net 時所得到的 performance 比使用從頭開始訓練的 ResNet-18 作爲 teacher net 時還要高，其原因大概是因爲 pretrained 並 fine-tune 過的 ResNet-18 其 performance 比從頭開始訓練的 ResNet-18 還要高，因此使用 pretrained 並 fine-tune 過的 ResNet-18 作爲 teacher net，可以提供 student net 更正確的 information，進而提高 student net 的 performance。

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

首先，我使用了助教所提供的 teacher net 和 student net 進行 knowledge distillation，其中，我使用了助教所提供的 data augmentation，並使用了 Adam 進行第一次的訓練，其中 batch size 爲 32，learning rate 爲 0.002，訓練了 150 個 epoch，接著，再使用 SGD 進行第二次的訓練，其中 batch size 爲 32，learning rate 爲 0.001，訓練了 50 個 epoch，以此得到 student net。接著，我將訓練好的 student net 進行 network pruning，其中，我使用的 compression ratio 分別爲 0.95 和 0.9，並進行了 5 次的 network pruning，在每次的 network pruning 之後，皆會使用 Adam 進行 fine-tune，其中 batch size 爲 32，learning rate 爲 0.001，並 fine-tune 了 5 個 epoch。network pruning 所得到的 validation accuracy 如下圖所示：



由上圖可以看出，當 compression ratio 越小，亦即 pruning rate 越大時，越不容易 fine-tune 回到原來的 accuracy，使得 accuracy 下降越多。

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

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

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

c. 將 CNN model 的 Convolution Layer 換成總參數量接近的 Group Convolution Layer (Group 數量自訂，但不要設爲 1 或 in_filters)。