

清華大學資工系大二

學號: A053095

林柏淵

- **Introduction:** Residual Network, 簡稱 ResNet(殘差網絡), 是 MSRA 何凱明團隊設計的一種網絡架構, 在 2015 年的 ILSVRC 和 COCO 上拿到了多項冠軍, 其發表的論文 Deep Residual Learning for Image Recognition, 是 CVPR 2016 的最佳論文。(來源於網路)

- **Experiment setup:**

Hyperparameter:

- Mini-batch size: 64 (782 iterations for each epoch)
- Total epochs: 164, momentum 0.9
- Initial learning rate 0.1, divide by 10 at 51, 75 epoch
- Weight decay = 0.0001
- self._make_layer1 = 30
- self._make_layer2 = 60
- self._make_layer3 = 90

detail of my model:

遇到比較大的問題在於, 因為助教要求 layer 需為 3 層, 而在 example code 中有 4 層, 所以這個問題的著重點就在於 trace code, 在把 resnet18 改成 resnet20 的同時將 4 layer->3 layer, 而最後我將 layer4 註解掉並由助教的 slide

Total Depth = 1 (conv) + 6n + 1(linear layer)

For example, to build ResNet-110, we need $n = 18$ $((110-2)/6)$.

Note that there is global average pooling before linear layer.

推測出 layer 的參數輸入方式應為[n,n,n]意即 layer 數的參數要相同, 而 resnet20 的參數為[3,3,3]依此類推, resnet56--n=9, resnet110—n=18,

而在 coding 方面由於 code 上已經有很多地方打好了我們要做的就是把參數修改加上自己的 code 以符合題目要求, 其中在執行方面, 我們在 run resnet110 時會有 out of memory 的問題, 而這時我發現兩種解決方案 1.將 plane 數縮小 2.將 batch size 縮小, 雖然事後證明縮小 batch size 是正確的選擇, 但將 filter 縮小也有相同效果, 可觀察到 filter 也有占一定的空間

```

self.conv1 = nn.Conv2d(3, 64, kernel_size=3, stride=1, padding=1, bias=False)
self.bn1 = nn.BatchNorm2d(64)
self.layer1 = self._make_layer(block, 64, num_blocks[0], stride=1)
self.layer2 = self._make_layer(block, 128, num_blocks[1], stride=2)
self.layer3 = self._make_layer(block, 256, num_blocks[2], stride=2)
self.layer4 = self._make_layer(block, 512, num_blocks[3], stride=2)
self.linear = nn.Linear(512*block.expansion, num_classes)

```

。

而執行的 performance 方面，發現 plane 的數量將與 training 時間呈正比但 plane 數量增加 Acc 也會跟著增加，原因推測在於 plane 增加會隨著將圖片中更多的 feature 記錄起來，讓之後的 classifier 預測更準確，但也隨著處理 plane 數增加而增加消耗時間，而同時也發現 batch size 縮小時(128->64)除了可以降低 memory 使用空間，也可以提升 performance，推測是由於 batch size 的減少導致每個 epoch 要 back-prop 的次數也隨之增加，所以也增加了梯度遞減的計算次數，因而讓 performance 更好，還有 slide 上提到的，learning rate decay 後來自己做了實驗，將 learning rate 分別在不同的 epoch 時間點除以 10(在 epoch50 開始調降，或 epoch90 開始 etc.)，得出的結論為當 Acc 已經在來回震盪的時候差不多就可以把 learning rate 調降。

而我也嘗試更換 optimizer function 像 example code 使用 SGD momentum Func.我把她更換成 adam function(URL:

https://pytorch.org/docs/stable/_modules/torch/optim/adam.html)

但最後 Acc 以 80 幾%的結果收場，我推測是由於 adam 的參數沒有設定好才導致這樣的結果，而另個原因是時間不夠，因為我是用 resnet110 來 test 所以 training 時間非常長，所以應該用更少的 resnet56 或 20 來 test 才對。

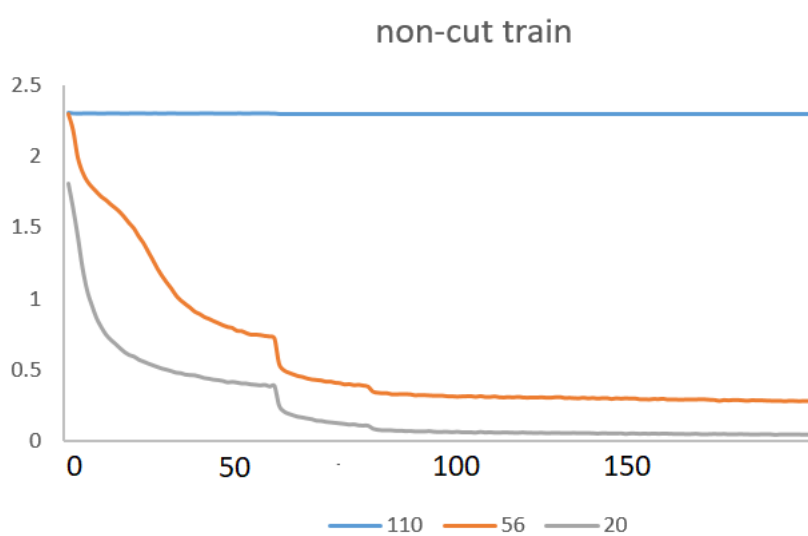
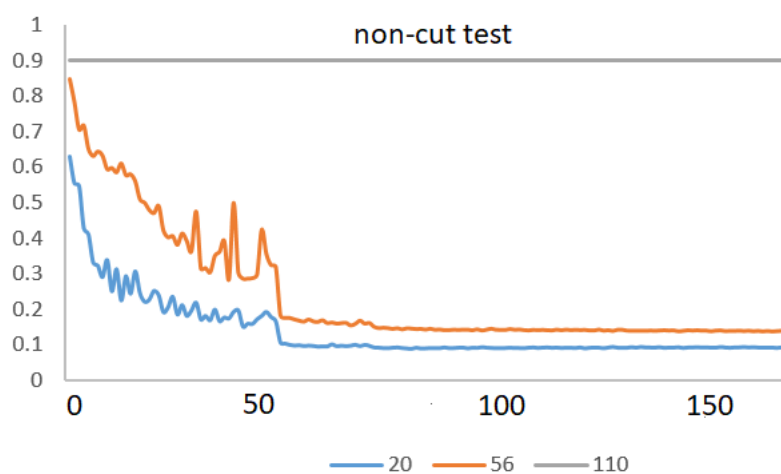
Final Test error:

Res20:100-93.110% = 6.890%

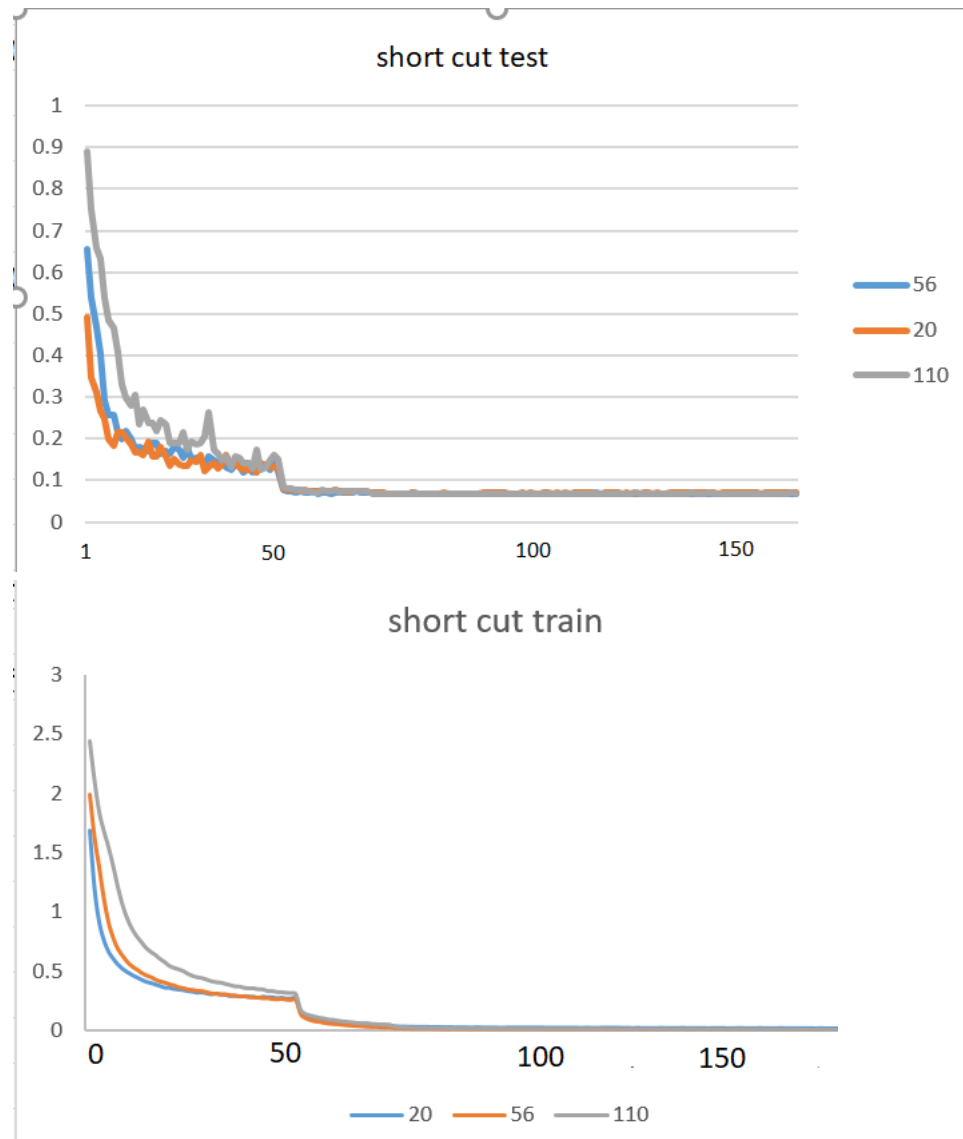
Res56:100-93.310% = 6.690%

Res110:100-93.430% = 6.570%

沒有 short-cut



有 short cut



- **Discussion:** 這是我上大學以來，第一次修這麼硬的課，英文授課、需要強大的數學基礎以及初次碰觸的語言 `python` 與要弄懂 `pytorch` 的語法，讓我這暑假過得相當“充實”，不過我沒有後悔來修這堂課就是了。

這次的 lab 學到很多 `python` 與 `pytorch` 函數與變數的運用，也看懂很多 `pytorch` 中 CNN 的 library 的運行方式，一開始把 code 下載下來後，一開始 run `main.py`

直接出現 error

..	
models	
LICENSE	1
main.py	4
README.md	1
utils.py	3

說到無法找到__ini__.py。。。。。。，大意是無法找到 model 目錄中的 resnet.py，後來索性把 resnet.py 拉到主目錄來問題就解決了。

