

Deep Learning and Practice - Lab1 report

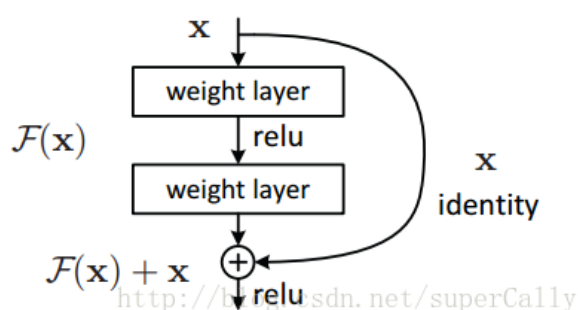
Introduction

Vanishing gradient problem

當網路逐漸加深時，前端的 Layer 的參數會因為 backpropagation 的計算(梯度連乘)以及 activation function 的設計(e.g. sigmoid)導致梯度消失 [1] [2]。

解決方式有使用 relu 作為 activation function，或是使用 Residual 的概念避免梯度消失。

Deep Residual Network

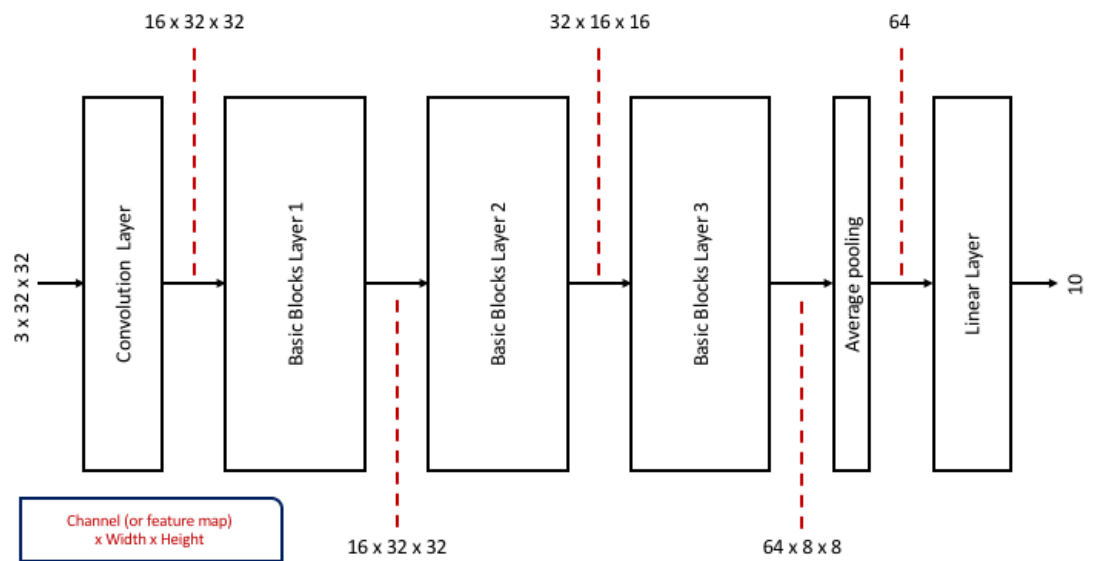


上圖為 DRN 中的一個 Basic block, 假設 Plain (w/o shortcut) 的網路輸出為 $H(x)$ ，也就是 $H(x) = F(x) + x$ ， $F(x)$ 即為把原值(x)去除掉後的殘差，由殘差的角度去討論，可以放大微小的梯度 [3]，如此一來可以避免梯度消失。

架構上第一層為一個 Convolution layer，接著為數層含有 Shortcut block Layer，最後進行 Average pooling 後接上 output Layer。

Experiment setup

1. Model detail



- **Basic Blocks Layer :**
包含了 3(Resnet20) / 9(Resnet56) / 18(Resnet110)個 Basic blocks。
- **Average pooling :**
Kernel size : 8×8 。
- **Linear Layer :**
Size : $64 \rightarrow 10$ 。

2. Parameters

- Method : SGD with momentum
- Momentum : 0.9
- Mini-batch size : 128
- Total epoch : 164
- Learning rate : Initial = 0.1, divide by 10 at 81th, 122th epoch
- Weight decay : 0.0001
- Weight initialization : `torch.nn.init.kaiming_normal`
- Loss function : cross-entropy
- For Data pre-processing :

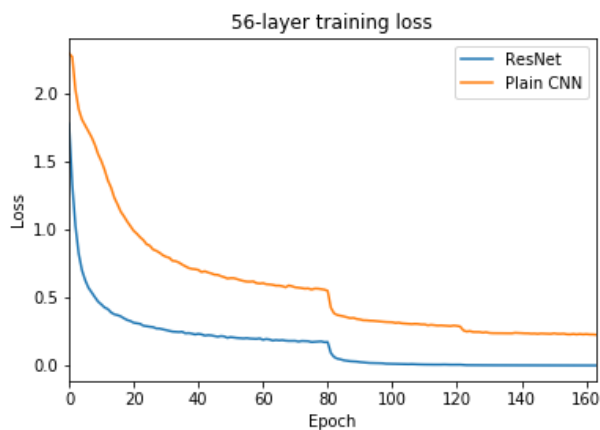
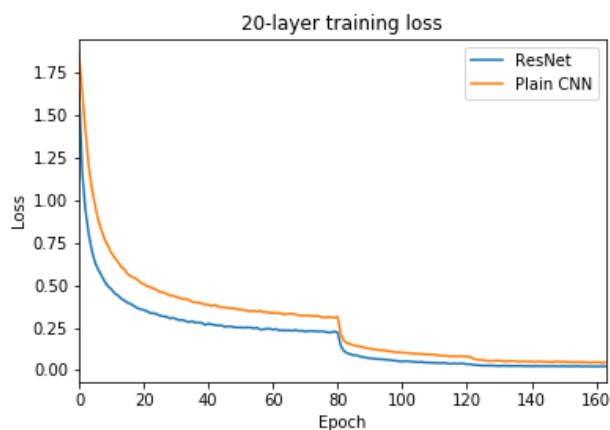
- ◆ Translation : Pad 4 zeros in each side and random cropping back to 32x32 size
- ◆ Horizontal flipping: With probability 0.5

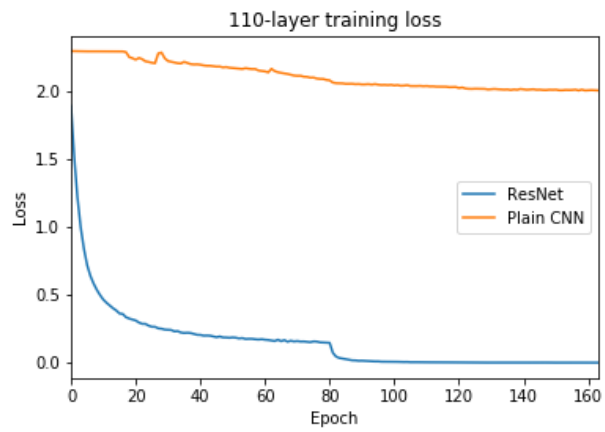
Result

Final test error rate

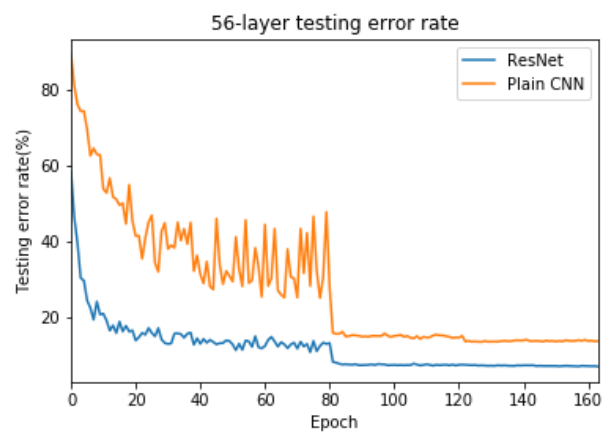
	20-layer	56-layer	110-layer
Plain CNN	9.93 %	13.68 %	75.17 %
<u>Resnet</u>	8.39 %	6.98 %	6.84 %

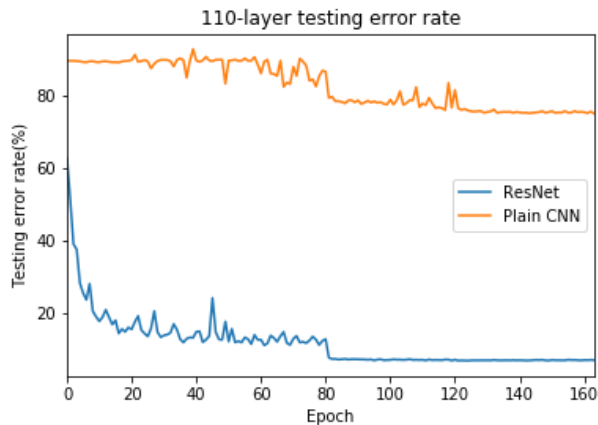
Loss curve





Test error curve





Discussion

由實驗結果的圖可以歸納出幾個結論：

1. Resnet 可以解決梯度消失的問題

當網路由淺至深時，可以發現 Plain CNN 在 train loss 以及 test error rate 的曲線都逐漸整個往上位移，在 layer-110 的 final test error rate 甚至高達 75 %；反觀 Resnet，兩個結果曲線受層數深淺的影響明顯不大，在 layer-110 時的 final test error rate 為最佳 (6.84%)。

2. 適時改變 Learning rate 對 performance 的影響

採用會變動的 learning rate，在第 81 以及第 122 epoch 時 learning rate 會乘上 0.1。由結果圖(loss 與 test error rate) 可以發現，在第 81 個 epoch 時，不管是 Resnet 或是 Plain CNN，皆有一個明顯的下降，而在第 122 個 epoch 則是部分的曲線有些微下降。在 learning rate 降低以前，loss 逐漸收斂以後在該範圍震盪，當 learning rate 降低的當下，參數的改變量也降低，因此可以透過移動更微小的步伐來跳脫出震盪的狀態，持續降低 loss。

Reference

- [1] <https://blog.csdn.net/supercally/article/details/55671064>
- [2] <http://wenda.chinahadoop.cn/question/6658>
- [3] <https://zhuanlan.zhihu.com/p/28413039>