

Method description

what is your strategy for parameter selection?

這次作業我選擇調整的因素有(optimizer 都是使用 Adam) :

1. 嘗試 resnet101、resnet152、vgg16 的 pre-training 模型
2. 不同的 batch-size、epoch、lr
3. 有無增加 data augmentation 的方式
4. 自己設計的模型 Net:
4 層 convolution、4 層 maxpooling、3 層 dropout、4 層 fully connection

Experience results

model	batch-size	epoch	lr	data augmentation	training accuracy	training loss	testing accuracy
vgg16	16	20	0.001		0.2929	2.9055	13%
Net	32	10	0.01		0.0232	4.8901	2%
resnet101	32	20	0.001		0.9500	0.1617	54%(hw2)
resnet101	64	200	0.0001		0.9978	0.0024	87%
resnet152	32	20	0.001	do	0.5736	1.6376	50%
resnet152	32	20	0.001		0.9341	0.2189	53%
resnet152	32	30	0.001		0.9627	0.1293	56%
resnet152	32	40	0.0001	do	0.9120	0.3399	80%
resnet152	32	40	0.0001		0.9979	0.0042	90%

training accuracy、training loss、testing accuracy 的圖片請使用 jupyter notebook 打開 code，即可看到。

Discussion

有比作業二的準確度高嗎?

結果討論與原因推論

實驗使用 vgg16 的準確率比作業二更低，我推論是因為 resnet101 是殘差網路(residual)，能減低梯度消失的問題，而 vgg16 沒有 residual。

我自己設計的 Net 模型，使用的層數遠比 pretraining 模型少，準確率也是遠低於作業二，可知網路層數多比較好。

resnet152 的模型層數比 resnet101 來的深，所以我預期 resnet152 的準確率會更高。而在 batch-size、epoch、lr 參數相同的情況中，使用 resnet152 的準確率並沒有比使用 resnet101(作業二)高。

但是在多次的實驗中，發現 resnet101 200 個 epoch 的準確率，比 resnet152 40 個 epoch 的準確率還低。也就是說，resnet152 以較少次的 epoch 得到較高的準確率。

在同為 20 個 epoch 的狀況中，resnet101 與 resnet152 的準確率差不多，但是更多次的 epoch 後，resnet152 的表現優於 resnet101。所以我推論，resnet101 訓練 200 個 epoch 可能產生梯度消失的問題，太多層數就會無法對前面網路層的參數進行有效的更新；resnet152 經過多次的 epoch 訓練後，就能找到比 resnet101 更準確的 accuracy。

另外，我也實驗了有無增加 transforms.RandomResizedCrop 和 transforms.RandomHorizontalFlip()的 data augmentation，transforms.RandomResizedCrop 是圖片的剪裁中心點隨機選取進行剪裁，transforms.RandomHorizontalFlip()是隨機水平翻轉圖片。實驗結果發現，有增加 data augmentation 並無法有效地使準確率提高。我推論是因為這次的資料集每張圖片中的車子都已經差不多在整張圖的中央了，而且車子頭在左在右對辨識影響不大，所以隨機水平翻轉、剪裁點隨機選取對這次的資料集幫助不大。

Problem and difficulties

在自己設計模型時，要考量到圖片大小經過 convolution、maxpooling 等等的轉變後，會變成的大小以及張數，使用如下圖的公式來計算。

Shape:

- Input: $(N, C_{in}, H_{in}, W_{in})$
- Output: $(N, C_{out}, H_{out}, W_{out})$ where

$$H_{out} = \left\lfloor \frac{H_{in} + 2 \times \text{padding}[0] - \text{dilation}[0] \times (\text{kernel_size}[0] - 1) - 1}{\text{stride}[0]} + 1 \right\rfloor$$
$$W_{out} = \left\lfloor \frac{W_{in} + 2 \times \text{padding}[1] - \text{dilation}[1] \times (\text{kernel_size}[1] - 1) - 1}{\text{stride}[1]} + 1 \right\rfloor$$

- Variables:**
- **weight** (*Tensor*) – the learnable weights of the module of shape (out_channels, in_channels, kernel_size[0], kernel_size[1])
 - **bias** (*Tensor*) – the learnable bias of the module of shape (out_channels)

並決定每層 fully connection 的節點數目，因為這些數字都是自己設計，沒有可以遵循的最佳參數規則，所以都是以試試看的實驗精神來做。