

Method description

第一個目標是要讀取 mat 檔中的數據，使用 `scipy.io.loadmat` 讀取 mat 檔的內容，得到 dict 格式(字典格式)的資料，再使用 `cars_annos['annotations']` 取出資料成矩陣格式，再將 bbox 的 x1, y1, x2, y2, class_id, 和圖片名稱一一取出並儲存進新定義的陣列中。

第二個目標是剪裁圖片並存到所屬類別名稱的資料夾中，使用 `opencv` 讀取、剪裁圖片，將 `cars_train` 資料夾中的圖片存到 `new` 資料夾中的 `train` 資料夾的各自所屬類別名稱的資料夾。完成資料的預處理。

第三個目標是依據作業指示，使用 `resnet101` 模型架構，並使用/不使用它的參數。使用 `model = models.resnet101(pretrained=True/False)` 載入模型，並使用

```
fc_features = model.fc.in_features
```

```
model.fc = nn.Linear(fc_features, 196)
```

將模型最後一層輸出類別數設定成這次資料集的 196 種。

How to run your test?

將資料解壓縮

打開 jupyter notebook

將助教放在 `ecourse` 的 `cars_test_annos_withlabels.mat` 檔案放進 `devkit` 資料夾中執行 `pre.ipynb`，完成資料預處理

執行 `10_pre_true.ipynb` 得到 `pretraines=true`、`epoch=10`、`lr=0.01`、`optim=SGD` 的結果

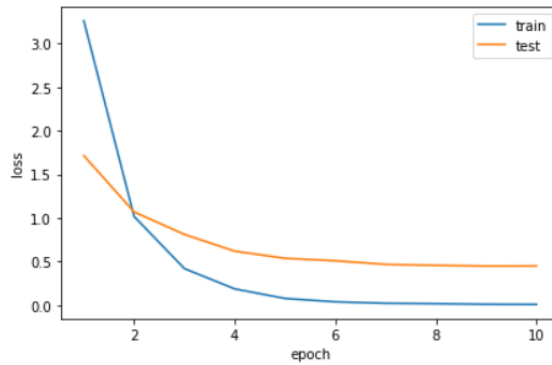
執行 `10_pre_false.ipynb` 得到 `pretraines=false`、`epoch=10`、`lr=0.01`、`optim=SGD` 的結果

執行 `20_pre_true-adam.ipynb` 得到 `pretraines=true`、`epoch=20`、`lr=0.001`、`optim=adam` 的結果

執行 `20_pre_false-adam.ipynb` 得到 `pretraines=false`、`epoch=20`、`lr=0.001`、`optim=adam` 的結果

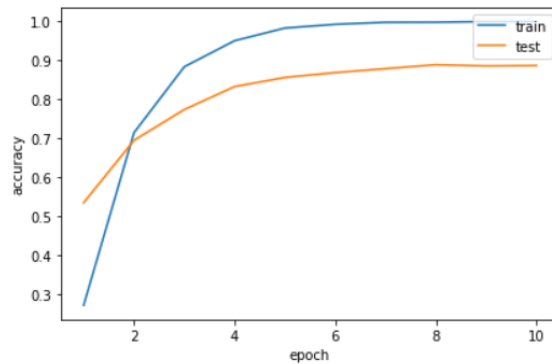
Experimental results

10_pre_true.ipynb

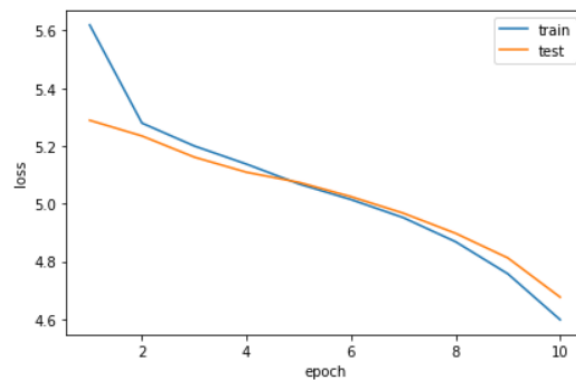


Accuracy on the ALL test images: 88 %

Accuracy of 0149 : 88 %
 Accuracy of 0025 : 92 %
 Accuracy of 0174 : 85 %
 Accuracy of 0151 : 81 %
 Accuracy of 0132 : 97 %
 Accuracy of 0010 : 84 %
 Accuracy of 0122 : 85 %
 Accuracy of 0143 : 92 %
 Accuracy of 0026 : 94 %
 Accuracy of 0146 : 97 %
 Accuracy of 0101 : 90 %
 Accuracy of 0075 : 77 %

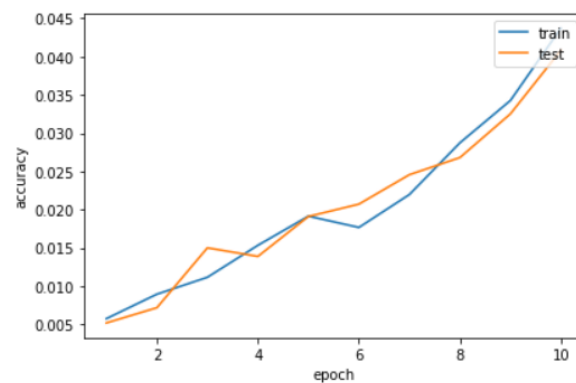


10_pre_false.ipynb

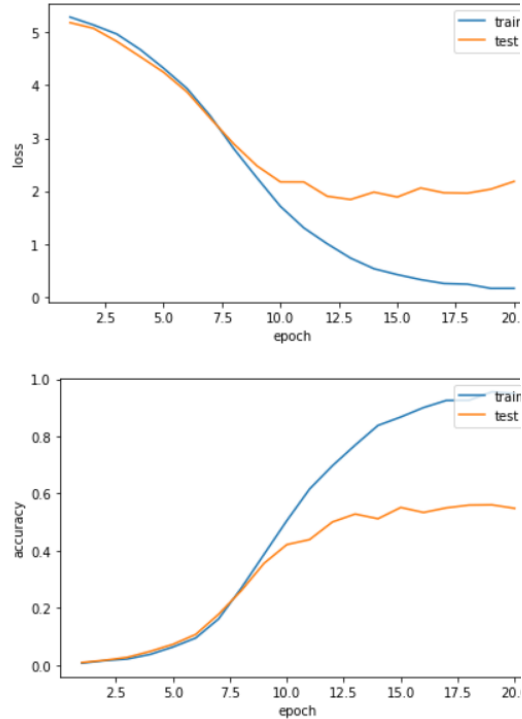


Accuracy on the ALL test images: 4 %

Accuracy of 0149 : 0 %
 Accuracy of 0025 : 0 %
 Accuracy of 0174 : 26 %
 Accuracy of 0151 : 0 %
 Accuracy of 0132 : 0 %
 Accuracy of 0010 : 0 %
 Accuracy of 0122 : 0 %
 Accuracy of 0143 : 0 %
 Accuracy of 0026 : 0 %
 Accuracy of 0146 : 0 %
 Accuracy of 0101 : 4 %

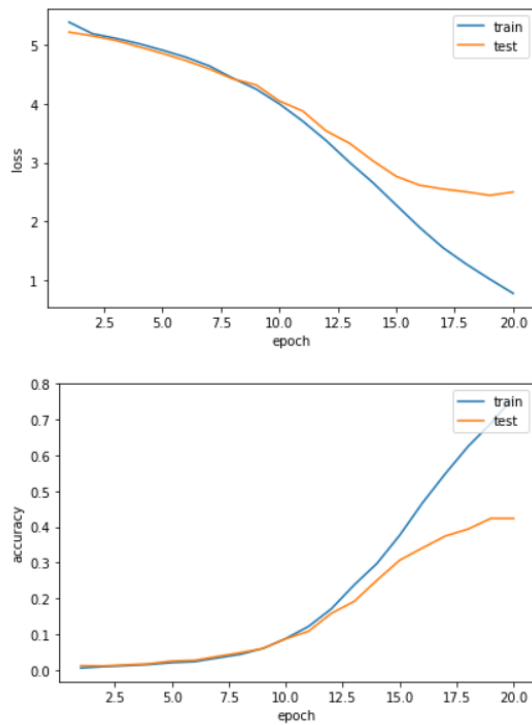


20_pre_true-adam.ipynb



Accuracy on the ALL test images: 54 %
 Accuracy of 0149 : 64 %
 Accuracy of 0025 : 74 %
 Accuracy of 0174 : 51 %
 Accuracy of 0151 : 55 %
 Accuracy of 0132 : 67 %
 Accuracy of 0010 : 60 %
 Accuracy of 0122 : 80 %
 Accuracy of 0143 : 65 %
 Accuracy of 0026 : 32 %
 Accuracy of 0146 : 72 %
 Accuracy of 0101 : 61 %
 Accuracy of 0075 : 36 %

20_pre_false-adam.ipynb



Accuracy on the ALL test images: 42 %
 Accuracy of 0149 : 71 %
 Accuracy of 0025 : 56 %
 Accuracy of 0174 : 9 %
 Accuracy of 0151 : 34 %
 Accuracy of 0132 : 46 %
 Accuracy of 0010 : 24 %
 Accuracy of 0122 : 55 %
 Accuracy of 0143 : 42 %
 Accuracy of 0026 : 26 %
 Accuracy of 0146 : 88 %
 Accuracy of 0101 : 54 %

Discussion

pretrained=true 時，準確率與上次的作業差不多。

pretrained=false 時，準確率大幅的下降，觀察 loss 的曲線變化，沒有明顯的過擬合現象，推測在資料數量不多、無使用預訓練模型的參數做訓練，準確率會不佳。

由於 pretrained=false 時的準確率非常不佳，所以我試試將優化器從 SGD 改成 Adam，epoch 再增加 10 個，lr 再除以 10 做測試。

這次 pretrained=true 時，準確率只有 54%，比優化器為 SGD 時差。

pretrained=false 時，準確率比優化器為 SGD 時佳，由此可推測優化器並沒有所謂的最佳優化器，而是要視情況而定。

觀察 pretrained=true 和 pretrained=false 的 accuracy 曲線變化，可以發現 pretrained=true 時 accuracy 上升速度較快，曲線呈現凹面往下；而 pretrained=false 時 accuracy 上升速度較慢，曲線呈現凹面往上。由此可知使用預訓練模型的參數有助於迅速提升準確率。

Problem and difficults

一開始很頭痛要怎麼讀取 mat 的數據並拿來剪裁圖片，幸好 github 上有人做過剪裁圖片可以參考，學到利用 `os.makedirs(dst_path)` 建立各個類別的資料夾、利用 `os.path.join` 將圖片放進指定的資料夾中。

訓練 pretrained=false 時，看到準確率非常低，覺得自己是不是沒寫對，和同學討論後，推測原因為這次的資料集每個類別大約只有 40 張，又沒使用預訓練模型的參數，所以準確率表現不佳。