

DeepLearning HW3

學號:313834004 姓名:周彥宏

Analyze data

這次的 Data 分成了 3 個 log file. 分別是 Training, Test, metrics. 三個 log file 的內容除了裡面的 feature 會有不同以外, 格式都是一樣的. 所以對於資料前處理可以使用通用的方法來去實作. 要注意的是每個檔案前 4 行是有關 Setting 的資訊, 這部分對於這次畫圖來說, 並不是需要抓取的, 所以會使用判斷是將前面的 Setting 捨棄掉.

Data preprocessing

```
if line.count('step') == 0:  
    continue
```

首先, 我想要一次讀取三個檔案, 所以把 Data 全部丟進同一資料夾中, import os module, 並使用 os.listdir function 來幫我抓取這個資料夾內所有檔案名稱, 變成一個 list. 接著我利用一個 Dictionary 來存三個 log file 的資料, 值得注意的是每個 log file 我也設成 dictionary 來存所有的 features.

如同剛剛上面所提到, 我在讀取檔案時, 是一行一行讀取, 然後判斷這行是不是有 step 來確定並不是前四行的 setting line.

```
pre_line = line.split('steps')  
steps = int(pre_line[1].split(':')[0])  
print(steps)  
if 'steps' not in data[filename]: # 避免沒有這個key in dict, 所以我做個判斷式來做創建  
    data[filename]['steps'] = []  
data[filename]['steps'].append(steps)  
all_features = pre_line[1].split('{')[1].split('}')[0].split(',')  
for feature in all_features[:-1]:  
    # print(feature)  
    name = feature.split(':')[0].strip().strip('\n') # 先刪掉頭尾空白, 再去刪掉頭尾單引號  
    feature_data = float(feature.split(':')[1])  
    # print(name)  
    if name not in data[filename]: # 避免沒有這個key in dict, 所以我做個判斷式來做創建  
        data[filename][name] = []  
    data[filename][name].append(feature_data)
```

接著我對要抓取的資料作前處理, 因為我們不需要 epoch 的資訊, 所以我選擇先用 steps 當作切分點把 epoch drop. 接著切分':' 來存 steps 的數值, 再切分左右大括號把所有 feature 切下來, 並觀察到中間都是使用逗號分隔, 所以又再使用逗號當切分點. 可以注意到我的 feature 只到[:-1], 是因為三份 log file,最後都會

多一個逗號, 且是空白. 我直接這樣做, 也不用去多做處理就捨棄他.
接著再讀取 feature name 時, 我使用到.strip()這個 function 來幫我去除 feature name 頭尾的單引號, 因為我是使用冒號當切分點, 把數據跟名稱分開.

Plot

```
plt.figure(figsize=(24, 18)) # 設定圖片大小
for i,ele in enumerate(data):
    plt.subplot(len(data), 1, i+1) # 在圖片中切data長度的分塊, 利用 i 來做分塊.
    title = f'{ele}'
    if ele != 'metrics':
        title += '_losses'
    plt.title(f'{title}')
    for feature in data[ele].keys(): # 除了steps以外的feature, 每個都依據steps當x軸, 畫出來
        if feature == 'steps':
            continue
        plt.plot(data[ele]['steps'], data[ele][feature], label=feature)
plt.grid() # 加上網格
plt.legend() # 加上圖例
plt.ylabel('value') # 設定y軸label
plt.xlabel('steps') # 設定x軸label
```

由於我想要一次性畫三張圖, 所以使用 subplot 幫我在整張畫布上切分成三個圖片. 接著我使用我處理好的資料來畫, 所有的 features(除了 steps), 都會將 steps 當成 x 軸參照後, 畫出線段. 並且給予 label(等於是在幫這條線命名), 這樣最後使用 plt.legend()時,才會顯示出來. 然後由於範例有給網格線, 所以我也有加上網格線.