
Machine Learning HW13

ML TAs

ntu-ml-2020spring-ta@googlegroups.com

作業內容

1. 實作 MAML 中的 regression 和 few-shot classification
2. 學習如何 improve MAML
3. 實作 Reptile

Omniglot 資料格式

omniglot.tar.gz

解壓縮後會有兩個資料夾: images_background (training set) 跟 images_evaluation (test set)

每個資料夾裡有不同文字, 像是下面的 Cyrillic.180, 180 是角度, 每個語言很多不同的字, 每個字有 20 個不同的 png 檔, 範例如下:

Omniglot/images_background/Cyrillic.180/character01/0218_06.png

原理

請自行參考老師的[投影片](#)

重要提示：

1. 一階 MAML: [投影片](#) p.25
2. 二階 MAML: [投影片](#) p.13 - p.18
3. [How to train your MAML?](#)

Report 1. (2.5%)

report 連結

: <https://reurl.cc/Qdn9z5>

配分: classification 修改 (1) regression 修改 (1) report 一階做法在 classification 上的 accuracy (0.5)

請描述如何將助教的程式碼 (包含 classification 與 regression) 從二階的 MAML 改成一階的 MAML (作答請盡可能詳盡, 以免助教誤會), 並且比較其最後在 classification 上的 accuracy (5-way-1-shot)。因此你的 GitHub 上會有 p1_classification.py 和 p1_regression.py 兩個檔案, 分別是 classification 和 regression 的一階版本。

Report 2. (2.5%)

pseudo code (1) 作圖(1) report accuracy (0.5)

請將 classification 的程式碼改成 inner loop 更新 5 次, inner loop 使用 adagrad 來更新參數。這題一樣要用二階的 MAML, 寫出其 pseudo code 與回報 accuracy (5-way-1-shot omniglot 分類任務)。並且以 outer loop 的更新次數為橫軸, 分類的準確率為縱軸作圖, 比較其差異。因此你的 GitHub 上要有 p2.py, 對應本題的程式碼。

Report 3. (2.5%)

配分: report 實作 tip 後的 accuracy (1) 解釋你使用的 tip (1) 找出助教實作的 tip (0.5)

實作論文 "How to train your MAML" (<https://arxiv.org/abs/1810.09502>) 中的一個 tip, 解釋你使用的 tip 並且比較其在 5-way-1-shot 的 omniglot 分類任務上的 accuracy。助教其實已經實作了一個, 請找出是哪一個 tip 並且不要重複。因此你的 GitHub 上要有 p3.py, 對應本題的程式碼。

Report 4. (2.5%)

配分: 程式碼 (2) 回報 accuracy (0.5)

請實作 reptile 在 omniglot dataset 上, 訓練 5-way-1-shot 的分類任務, 並且回報其 accuracy。這題應該在 GitHub 上會有 reptile_train.sh 與 reptile_test.sh 的 shell script, 分別對應 meta training 與 meta testing。

老師投影片 p.30: [http://speech.ee.ntu.edu.tw/~tlkagk/courses/ML_2019/Lecture/Meta1%20\(v6\).pdf](http://speech.ee.ntu.edu.tw/~tlkagk/courses/ML_2019/Lecture/Meta1%20(v6).pdf)

本題助教會跑 **Training** 的 code, 請不要讓你的 code (在 colab 上用 GPU) 跑超過 4 小時。

第四題程式碼規定

reptile_train.sh請實作meta_training, 並將train完的model命名為model_學號.bin
reptile_test.sh請實作將上述存起來的model進行meta testing

助教在執行時會以以下方式執行：

```
bash reptile_train.sh <PATH_TO_OMNIGLOT_train_set>  
<PATH_TO_MODEL>
```

```
bash reptile_test.sh <PATH_TO_OMNIGLOT_test_set> <PATH_TO_MODEL>
```

- PATH_TO_OMNIGLOT 是 Omniglot 的 directory, 像是在助教的 code 是 ./
- PATH_TO_MODEL 是訓練之後產生的模型檔案儲存位置, 模型名稱請取名為 model.bin, 所以如果 PATH_TO_MODEL 是 /home/usr/model_dir, 助教執行完 reptile_train.sh 之後就應該要在 /home/usr/model_dir 裡面儲存一個 model.bin 的模型檔案。

作業注意事項

- 每一題都應該有對應的程式碼，助教在評分時會以 report 的敘述為主，如果助教認為你的正確率很可疑或是敘述我們看不懂，我們會去看你的程式碼。如果我們看不懂程式碼或是程式碼是錯的，那該題就不會有分數。
- 特別注意第 4 題助教會實際執行 **training** 程式碼。
- 請不要把 model 參數直接寫死在 reptile_train.sh 裡面，被發現直接 0 分

GitHub Submission Format

GitHub 的 hw13-<account> 上應該至少會有以下檔案, pX 對應到的就是 Problem X:

1. p1_classification.py
2. p1_regression.py
3. p2.py
4. p3.py
5. reptile_train.sh
6. reptile_test.sh
7. report.pdf

注意事項

- Omniglot 的 accuracy 請回報 5-way 1-shot 的 **TESTING ACCURACY**(程式碼最後一行 print 出來的東西), 不是 training accuracy。
- 請不要在 reptile_train.sh與reptile_test.sh 中把 omniglot 這個資料夾以及test時的model路徑寫死, 如果路徑寫死導致助教無法執行, 該題 0 分。
- 有用到 path 的部分, 請使用 os.path.join()
- 有任何問題, 可以在 TA hour 來問, 或是可以寄信到[課程信箱](#)。請勿私訊助教, 助教只會封鎖你。