
Machine Learning HW9

ML TAs

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

Image clustering - outline ^{1/6}

- 目標:分辨給定的兩張 images 是否為風景 (植物也算風景 e.g., 一片葉子)。
 - 除了 image 都是 $32*32*3$ 的圖片, 沒有任何 label
 - 只能用我們給的 data, 不能使用額外的 dataset , 也不能使用額外資料train 的 model



V.S



Image clustering - data 2/6

- 請同學以 `np.load()` 讀入資料, `valX.npy` 和 `valY.npy` 只用來檢驗我們的訓練效果, **不能用來訓練**。
- `trainX.npy`
 - 裡面總共有 8500 張 RGB 圖片, 大小都是 $32 \times 32 \times 3$
 - `shape` 為 `(8500, 32, 32, 3)`
- `valX.npy`
 - **請不要用來訓練**
 - 裡面總共有 500 張 RGB 圖片, 大小都是 $32 \times 32 \times 3$
 - `shape` 為 `(500, 32, 32, 3)`
- `valY.npy`
 - **請不要用來訓練**
 - 對應 `valX.npy` 的 `label`
 - `shape` 為 `(500,)`

Image clustering - methods ^{3/6}

- 如果直接在原本的 image 上做 cluster, 結果會很差 (有很多冗餘資訊)

→ 需要更好的方式來表示原本的image

- 為了找出這個更好的方式, 可以先將原始 image 做 dimension reduction, 用比較少的維度來描述一張 image

e.g., autoencoder, PCA, SVD, t-SNE.

Image clustering - requirements 4/6

1. 請實作用 **autoencoder** 將 8500 張圖片降維
2. 再利用降維過的 latent code 做分類
3. 預測 8500 筆測資是否來自相同的 dataset

註：實作的方法需含有 autoencoder，但還是可以將其他的降維方法一起搭配使用

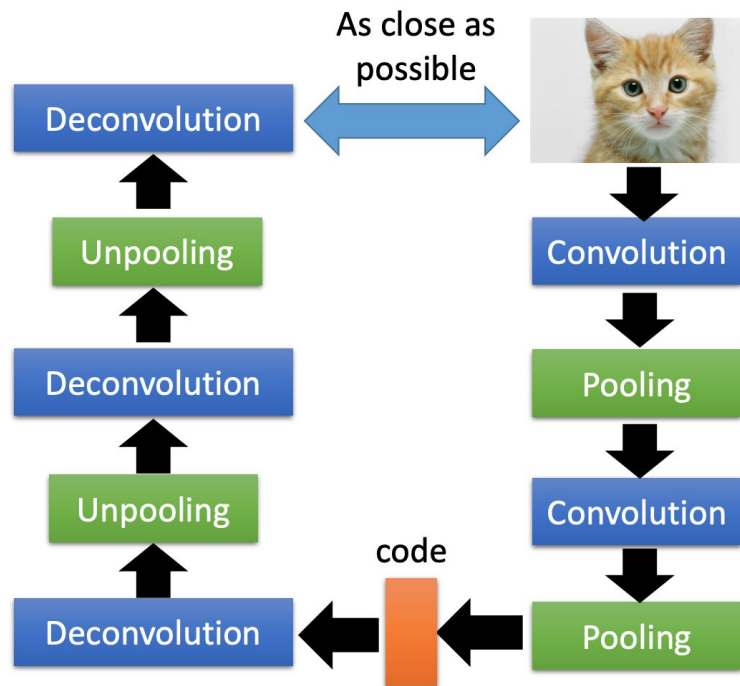


Image clustering - methods (cont.) 5/6

- 接著對降維過後的數據做 cluster
 - 可以試試 K-means
- 或者你可以衡量兩個降維過後的 images, 他們之間的相似度 (similarity)。如果相似度大於一個設定好的 threshold, 就把這兩個 images 當成同一類別
 - 算 similarity 的方法: euclidean distance, cosine similarity.....

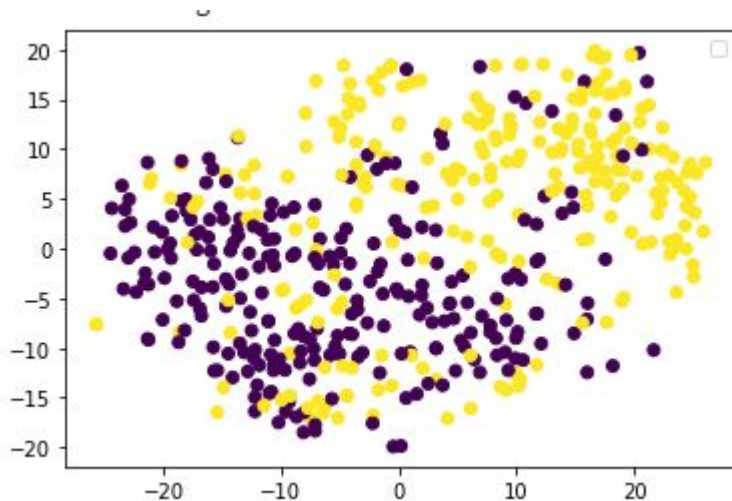
Image clustering - methods (cont.) 6/6

- 其他可能有幫助的事：
 - 必須找個方法來衡量方法的好壞，一個直覺的方法是利用降維過後的feature 去 reconstruct 成原本的 image。如果 reconstruct 的結果越接近原本的 image，可以一定程度的代表你抽出來的 feature 越好
 - 對原始 image 做 data augmentation
 - 二次降維
 - SOTA Unsupervised Learning: MoCo, SimCLR (有興趣的同學可以去看論文)
 - 看看網路上的 unsupervised learning 內容
 - <https://stanford.edu/~shervine/teaching/cs-229/cheatsheet-unsupervised-learning>
 - <http://www.mit.edu/~9.54/fall14/slides/Class13.pdf>
 - <https://sites.google.com/view/berkeley-cs294-158-sp20/home>

Report 1/3

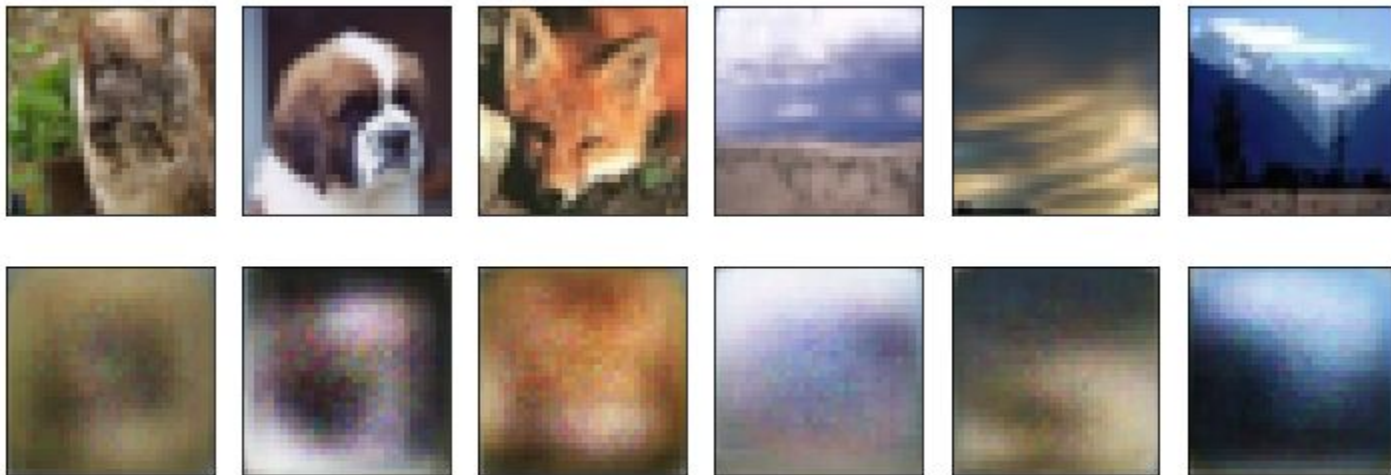
- (3%) 請至少使用兩種方法 (autoencoder 架構、optimizer、data preprocessing、後續降維方法、clustering 算法等等) 來改進 baseline code 的 accuracy。
 - 記錄改進前、後的 accuracy 分別為多少。
 - 使用改進前、後的方法, 分別將 val data 的降維結果 (embedding) 與他們對應的 label 畫出來。
 - 盡量詳細說明你做了哪些改進。

作圖範例, report 中需要畫兩張圖:
baseline model、improved model 各一張



Report _{2/3}

- (1%) 使用你 accuracy 最高的 autoencoder, 從 trainX 中, 取出 index 1, 2, 3, 6, 7, 9 這 6 張圖片。
 - 畫出他們的原圖以及 reconstruct 之後的圖片。



作圖範例

Report 3/3

- (2%) 在 autoencoder 的訓練過程中, 至少挑選 10 個 checkpoints。
 - 請用 model 的 reconstruction error (用所有的 trainX 計算 MSE) 和 val accuracy 對那些 checkpoints 作圖。
 - 簡單說明你觀察到的現象。

作圖範例, 同學最好另外在橫軸標出單位, 例如這幾個點是在第幾個 steps (或是 epochs) 儲存的 checkpoint



Submission ^{1/2}

- report.pdf
- *.py
 - 請上傳你所有會需要用到的 .py 檔
 - ex: train_baseline.py, train_improved, model.py
- checkpoints/baseline.pth
 - 你使用在 Report Problem 1 的 baseline model
- checkpoints/improved.pth
 - 你使用在 Report Problem 1 的 improved model
- checkpoints/best.pth
 - 你在 Kaggle 上面最高分的 model, 也是 Report Problem 2 的 model (可以和 checkpoints/improved.pth 一樣)

Submission 2/2

- hw9_best.sh
 - 用同學繳交上來的 checkpoints/best.pth 執行testing, 也就是降維+預測的部分, 要跟kaggle上你的最高分數差不多
 - **用法: `bash hw9_best.sh <trainX_npy> <checkpoint> <prediction_path>`**
 - trainX_npy: 助教這邊存放 trainX.npy 的路徑, 請同學不要寫死
 - checkpoint: 助教這邊存放 model 的路徑, 請同學不要寫死
 - prediction_path: 助教這邊存放 prediction 的路徑, 請同學不要寫死
 - 範例:
 - `bash hw9_best.sh ~/data/trainX.npy ~/chekpoints/best.pth ~/saved/prediction.csv`
- train_baseline.sh
 - 說明: 訓練 Report Problem 1 的 baseline model, 可以 reproduce 同學繳交上來的 checkpoints/baseline.pth
 - **用法: `bash train_baseline.sh <trainX_npy> <checkpoint>`**
 - trainX_npy: 助教這邊存放 trainX.npy 的路徑, 請同學不要寫死
 - checkpoint: 訓練完 model 之後要存檔的路徑, 請同學不要寫死
 - 範例:
 - `bash train_baseline.sh ~/data/trainX.npy ~/chekpoints/baseline.pth`
- train_improved.sh
 - 用法同上, 能 reproduce 同學繳交上來的 checkpoints/improved.pth
- train_best.sh
 - 用法同上, 能 reproduce 同學繳交上來的 checkpoints/best.pth

Reproduce Regulation

- 請務必在訓練過程中，隨時存取參數。
- 請同學確保你上傳的程式所產生的結果，會跟你在 Kaggle 上的結果一致，基本上誤差在 ± 0.03 之間都屬於一致，若超過以上範圍，Kaggle 將不予計分。
- Testing執行時間上限為 **10** 分鐘。
- Training執行時間上限為 **30** 分鐘。

Links

- Kaggle: <https://www.kaggle.com/c/ml2020spring-hw9>
- Colab: <https://reurl.cc/Qdn7Mo>
- Report template: <https://reurl.cc/O17XO9>
- 遲交表單: <https://bit.ly/39d2x2m>