

Machine Learning 2016

系級：電機四

姓名：鍾勝隆

學號：B02901001

HW3 Report – Semi-supervised Learning in Picture Classification

1. Describe supervise learning

一開始的 5000 筆 label 的資料，先經過簡單的 numpy.array 的 parsing，將維度轉成 (5000, 3, 32, 32) 的形式，除以 255，將數值都調整至 0 到 1 之間，並產生對應的 label (length 為 10 的 array，對應的 data label 位置為 1，其餘為 0)，label 維度為 (5000, 10)，完成初步的資料前處理。使用 keras 建立 Sequential 的 model layer，循環使用：

Convolution2D → Activation → MaxPooling2D

做圖像處理，再使用 Flatten 攤平，輸入 Fully Connected 的 neural network，最後使用 Categorical_crossentropy 計算 loss，並以 Adam 作為 optimizer。

因為考慮到測資有點少，我有使用 ImageDataGenerator 產生更多有效測資，將圖片做微幅旋轉、平移，使得 neural network 可以對應更對不同的資料。另外，由於需要確認 Training 的 fitting 程度，使用 validation_split，切出部份的資料以做評估，設定 EarlyStopping 監控，一旦 val_loss 不再連續下降到一定次數，就將程式停下。

2. Describe semi-supervised learning (method1)

在 semi-supervised learning 中，我主要是以 selfing-training 的方式實作，需要評估該加哪些 unlabel 的檔案進去 training 才会有更好的結果，先以 label 的資料，輸入進 neural network，再 predict unlabel 的資料，由出來的結果判斷，該 unlabel 資料最高機率的類別必須大於 0.8，才會將該資料加入下一次的 training data，再將 model 進行一次 training。反覆的進行前述的動作，直到取出一定量的 unlabel data。

但是由於觀察到，如果特定的資料種類過多，在 predict 時，也會偏向將 data predict 成該種類，所以，我特別在每次增加資料時，使每個種類增加了量都會相同，使得 training set 的資料較為平均，predict 的資料也不會過於傾向某一種資料種類，但是這樣每次增加的 unlabel 資料就有限，因為必須是最低可以增加的數量，有的種類可能

無法增加太多（predict 出來較少）。取得較好的準確度。

由於 keras 輸出 model 的格式有特別要求（副檔名須為 h5），所以，test.sh 需要在 model name 後加 ‘.h5’，以成功讀入 model。

3. Describe semi-supervised learning (method2)

在 method2，我是實作 autoencoder 與 cnn 的 self-training，由於處理的資料是圖像，我使用 convolutional autoencoder，透過連續的 convolution layer，將原來的資料 mapping 到維度較小的 feature data。Training autoencoder 時，我是使用 Binary_crossentropy 作為 loss，Adam 為 optimizer。train 完後，將 autoencoder 的 encoder 取出，把資料都轉成 feature data，再丟入先前的 self-training 的 cNN。

因為同時有使用兩種 model，所以 train.sh 和 test.sh 都需要多一個 model 的參數，以可以正確地將 model_semi_super 以及 model_autoencoder 輸出或讀入。並也同前面所述，test.sh 需要在 model name 後加 ‘.h5’，以成功讀入 model。

儘管更改完 model 的深度和廣度，autoencoder 的 loss 一直無法有效下降，這次並沒有很成功的實作出來較好的結果。

4. Reference

NTUEE – ML2016Fall 上課投影片

<http://speech.ee.ntu.edu.tw/~tlkagk/courses ML16.html>

Keras Documentation

<https://keras.io/>

Autoencoder

<https://blog.keras.io/building-autoencoders-in-keras.html>