ML2016 HW3 Report

R05921040 謝友恆

- Objective

  給定 labeled data 5000 筆與 unlabeled data 45000 筆，使用 semi-supervised 的方式預測 data 所屬的 class。

- Data Preprocessing: Data augmentation

  Data augmentation 指對圖像進行一些變換，例如平移、旋轉、縮放以及加上雜訊等等，並將經過這些變換的圖像當作新的資料去訓練。使用這個方式可以降低雜訊對資料的影響，讓輸出的模型可以辨識更廣泛的資料。
  本次作業中使用 Keras 中的 ImageDataGenerator 達到這個效果。

- Supervised Learning

  Supervised learning 與 Unsupervised learning 最大的差別就是給定了各筆資料所屬的 class，因此在訓練的過程中可以得知在 training data 上實際的 error。

  - Feature Extraction

    Convolution(32, 3, 3) * 2 + Maxpooling(2, 2) + Dropout(0.25) +
    Convolution(64, 3, 3) * 2 + Maxpooling(2, 2) + Dropout(0.25) + Flatten()
    最終得到一個 64 * 8 * 8 的矩陣，Flatten 之後為 4096 維。

  - Model

    Dense(512, activation = “relu”) + Dense(10, activation = “softmax”)

  使用原始 data 可以得到 56%的準確度，已經較 baseline 好過不少。而使用 100 epoch 的結果會較使用 40 epoch 的結果好大約 3 個百分點，而使用 200 epoch 的結果則跟 100 的結果差不多。
  而若是加上 data augmentation 則可以達到 67%的準確度，由此可以 data augmentation 對於 image classification 有不錯的效果。

- Semi-Supervised Learning

  - Method 1: Self Training

  Self Training 的部分依據作業投影片，以 supervised learning 得到的 model 去預測 unlabeled data，並在每一個 iteration 中取其中 confidence 最高的加上 label 重新預測 model，一直到固定的 iteration 數量或是 labeled data 超過一定數量為止。此處使用的 model 與 supervised 相同。Confidence 門檻設為 0.999，達到 iteration 15 次或是 labeled data 超過 20000 筆則停止。
  使用這個方法加上 data augmentation 可以達到本次實驗中最好的結果。有接近 73%的正確率，而若不使用 data augmentation，也較 supervised 最好的結果好一點。參數的部分，confidence 由 0.99 提升至 0.999 與 0.9999 都有增進效能，而 0.9999 卻又較 0.999 差了一些，或許是因為有些已經足夠準確的資料被浪費掉了。而使用 100 epochs 會比 40 epochs 好上約 5 個百

分點，推測尚未 overfit，但用到 200 epoch 是否會讓表現變得更好則需要更多的實驗，可惜因為時間因素無法達成。

- Method2: autoencoder clustering

這個方法我們使用 autoencoder 先從資料抽取 256 維的特徵，計算每個 class 的特徵平均值，並用 unlabeled data 中每一筆資料的特徵與各 class 特徵的 l2 norm 來標記一些 unlabeled data。最後使用這些 data 再去 training。然而使用這個方式結果卻不甚理想，就算加上 data augmentation 也只有 56%的準確度，與 supervised learning 相比準確度下降了 10 個百分點。這應該是因為 l2 norm 的距離並沒有表現出資料間真實的情況，或許使用 clustering 技巧如 knn 或 kmeans 會有比較好的效果。

- Discussion:
  - Performance:

在本次的三個做法之中，self training 可以達到最好的效果，我覺得這次我的實驗還沒有完全發揮 self training 的能力，如果再改動 epoch 的數量、confidence 或是 labeled data 的數量門檻應該還可以達到更好的結果，然而卻也會需要更多的時間。本次使用的機器是 i7 6700k + GTX 970，而目前最好的 100 epoch + 0.999 confidence + number of labeled data 20000 的執行時間約為半小時，若是提高 epoch 數量或 data 數量門檻，則執行時間可能會增加數倍。

Model 複雜度與效能的 trade off 也是之後實際應用時必須考量的重點，如果花了雙倍的時間與空間卻只得到 10%的改進，則應該要視應用種類決定是否要採用這樣的方式。現今熱門的 NN 壓縮便是犧牲了網路的準確性以達到更小的 model 以及更快的計算速度，才能運用在計算能力不強的行動裝置上。

- Autoencoder:

本次實驗原本 Autoencoder 的 input 是沒有經過修改的 image data，然而無論是使用單層、多層，甚至是 convolution 的模型都得到十分糟糕的結果。之後用與 supervised learning 相同的 feature extraction 先抽取出部分 feature 再放進 autoencoder，對於效能就有顯著的改善。推測是因為 raw data 異質性太高，而 feature extraction 可以將其中較有特色的部分抽取出來，因此自然有比較好的效果。

- Data Augmentation:

本次 data augmentation 參數主要參照網路上的教學文章設置，只有對資料做小範圍旋轉、平移與翻轉，可以考慮加上其他參數，如 zca whitening、normalization 等等，或許會有更好的結果。