Machine Learning
HW3

r05922018
黃柏智

1. Supervised learning

我採用的方法是 convolutional neural network，使用 Keras 這個 package 來實作。
在進入 CNN 之前，沒有對原始的 image 做任何處理，並使用 SGD 來優化 CNN。
這個 SGD 包含了 weight decay 以及 momentum。
我這個 CNN 是由 4 層 convolutional layer 以及 2 層 fully-connected layer 組成，除了 output layer 使用 softmax 來當 activation 以外，其餘的 activation 都使用 ReLU (rectified linear unit)。
以下是實驗中得出的經驗：

- 經過數次試驗後發現，在做 SGD 時，如果 batch-size 太小，準確率會停滯不前，batch-size 至少要 64 以上，準確率才會隨著 epoch 提升。最後設定的 batch-size 為 100。
- 此外，dropout 也是個不可或缺的方法。不加 dropout，準確率會幾乎跟 baseline 一樣，加了之後，準確率至少會提升 5%。dropout 的大小最後訂在 0.55，高過或低過這個數字所得到的準確率都比較低。
- 除了 SGD 之外，我也嘗試用 adam, adadelta, nadam 來做優化，但最後的準確率都比較低。
- Keras 的 convolutional layer 中有一個調整 zero_padding 的參數 'border_mode'，最後得出設定成 'same' 會比設定成 'valid' 好。

2. Semi-supervised learning(1)

第一種 semi-supervised learning 的方法是 self-training。使用 CNN train 出來的 model 來對 unlabeled data 做預測，然後使用 labeled data + unlabeled data 重新 train 一次 CNN。用 train 好的 model 來預測 unlabeled data，並使用它來重 train model 的步驟會重複數次。而 CNN 的參數跟第一題中的 CNN 一模一樣。
經過實驗後發現，self-training 的結果會比原本更差。或許是因為 CNN train 出的 model 準確率還不夠高，也可能是因為 unlabeled data 的數量比 labeled data 多出太多。而且把 unlabeled data 加入 training data 的步驟重複越多次，準確率會越來越低。
如果對 unlabeled data 沒有很好的預測，等同於整個 training data 有很多錯誤 label，用這個 training data 下去 train CNN，會把整個 model 帶往不好的方向，所以準確率才會不升反降。
我也有試著只加入 unlabeled data 中，經過 model predict 後機率大於某個值的 data。使用這些 data，結果還是會降，但降幅變小許多。這個值大約介在 0.8~0.9 之間。

3. Semi-supervised learning(2)

第二種 semi-supervised learning 的方法是使用 autoencoder。autoencoder 會把原始的 image encode 成一串 code，再試著利用這些 code 去還原成原始的 image。我試過 NN 以及 CNN 來做 autoencoder 的架構，發現用 CNN 來做，image 的還原度較高。
整個 CNN 有四層 convolutional layer，encoder 與 decoder 各有兩層。而優化 CNN 的方法是 adam。
以下是實驗中得出的經驗：

- 我試了幾種優化 CNN 的方式，其中 adam 的表現勝過 adadelta, SGD, nadam，而且 adam 收斂的速度非常快。
- data 越多，做出來的 autoencoder 效果越好。因此我使用了 labeled data + unlabeled data + test data。
- 加上部分噪音後，train 出來的 autoencoder 會更加 generic

在 train 完 auto encoder 後，必須用這個模型來對 unlabeled data 做分類。
我嘗試了 k-means 與 k nearest neighbors (KNN) 兩種分類方法，並使用 sklearn 這個 package 來實作。
在做 k-means 時，必須給定每一個類別的初始中心點。我的算法是先把 labeled data 丟進 encoder，並用 encode 出來 code，分別算出 10 個類別的平均值。平均值的算法就是直接用 np.mean 來計算 encode 出來的 code (code 是個 ndarray)。
至於 KNN，一樣是用 labeled data encode 過後的 code 下去 train。在對 unlabeled data 做預測時，會取 7 個最相似的 neighbors，並依照 euclidean_distance 來計算每個 neighbor 的權重。
最後的結果，KNN 的準確率會比 k-means 要高，但準確率還是很低。在無法對 unlabeled data 做好的預測的情況下，仍用 unlabeled data 來重 train 一次 CNN，可想而知準確率會下跌，而我做出來的結果也是如此，即使有設一個 KNN predict 機率的 threshold，超過 threshold 的 data 才拿來用，準確率還是會下降。

4. Compare and analyze your results

在上面三種方法中，成果最好的是只用 labeled data 來做 image classification。
用了 self-training，以及使用 autoencoder + KNN，都無法使結果提升。
self-training 不能提升準確率的原因，可能是一開始的 CNN 就沒有很高的準確率，用這個不高的準確率，來預測數量是 labeled data 九倍的 unlabeled data，會讓整個 training data 品質下降，進而導致準確率不升反降。
而 autoencoder 不能提升準確率的原因比較難確定。原因可能出在 KNN 這段，也可能出在 autoencoder 本身，畢竟優化 autoencoder 也是一個難題。當 clustering 的準確度不高時，就會發生跟 self-training 一樣的問題，也就是 training data 的品質下降，導致準確率也下降。
不過在數次實驗中也發現，使用 encoder 過的 data 來做 clustering，其準確度的確優於不做 encoding 的 data。這證明了 autoencoder 真的有抽取出 image 的一些特性。