

1. Supervised Learning

我嘗試了非常多種CNN結構，調整過深度、寬度、dropout、pooling method、border_mode等，最後採用了一個在速度與準確度達到良好平衡的model，其結構如下：

- Convolution layers
 - Convolution2D(64, 3, 3, border_mode='same')
 - Convolution2D(64, 3, 3, border_mode='same')
 - MaxPooling2D(pool_size=(2, 2))
 - Dropout(0.5)
 - Convolution2D(64, 3, 3, border_mode='same')
 - Convolution2D(64, 3, 3, border_mode='same')
 - AveragePooling2D(pool_size=(2, 2), border_mode='same')
 - Dropout(0.5)
- Fully-connected layers
 - Dense(512), Dropout(0.5)
 - Dense(256), Dropout(0.5)
 - Dense(128), Dropout(0.5)
 - Dense(10), Activation('softmax')











這個model可以在100 epoch達到training accuracy 0.92、testing accuracy 0.6276。

2. Semi-supervised learning (1)

- 我使用的方法是Self-training，model的結構採用前述的CNN結構。
- 在用labelled data train出第一個model並用這個model來標記unlabeled data後，我會根據predict的機率分佈，抽出entropy最高的前40% unlabeled data加到labelled dataset中，再利用這個labelled dataset train CNN。
- 我測試過抽取10%、20%、30%、40%、50%、60%，發現40%及50%的效果最好，可以達到0.66左右的test accuracy，60%以上可能因為標記錯誤的資料變多了，會害accuracy下降。
- repeat的次數我也測試過，結果發現repeat第二次之後的accuracy都沒有明顯提升，所以最後使用的方法只做了一次「抽取unlabeled data加到labelled dataset」的動作。
- 使用這個方法得到的就是我的最佳結果：private score 0.6674

3. Semi-supervised learning (2)

- 我使用的第二個Semi-supervised learning method是autoencoder。
- 我並沒有做clustering，而是把autoencoder作為pre-trained CNN model來用。
- 訓練步驟如下：
 - 用 labelled data + unlabeled data train autoencoder
 - 從autoencoder取出encoder的前面幾層，當作pre-trained CNN，對labelled data 做predict可以得到一組向量
 - 把這些向量當作input，labelled data的label當作output，train 一個 DNN當作classifier
 - 如此一來encoder + classifier就組成了一個image classifier
- autoencoder的結構如下：
 - Encoder (output = 16*8*8 tensor)

- Convolution2D(64, 3, 3, border_mode='same')
 - Convolution2D(64, 3, 3, border_mode='same')
 - MaxPooling2D(pool_size=(2, 2))
 - Convolution2D(64, 3, 3, border_mode='same')
 - Convolution2D(64, 3, 3, border_mode='same')
 - AveragePooling2D(pool_size=(2, 2), border_mode='same')
 - Convolution2D(32, 3, 3, border_mode='same')
 - Convolution2D(16, 3, 3, border_mode='same')
 - Convolution2D(8, 3, 3, border_mode='same')
 - AveragePooling2D(pool_size=(2, 2))
- Decoder
 - 結構跟Encoder相反，就不贅述了。
- DNN classifier的結構如下：
 - Dense(512), Dropout(0.5)
 - Dense(256), Dropout(0.5)
 - Dense(128), Dropout(0.5)
 - Dense(10), Activation('softmax')
- 結果
 - autoencoder
 - | | | | | | |
|----------|---|---|--|---|---|
| original |  |  |  |  |  |
| decoded |  |  |  |  |  |
 - loss = 0.5522
 - 可以看出雖然資料量被壓縮到1/6，但除了細節變模糊外，還原的效果還不錯
 - encoder + classifier
 - test accuracy = 0.3654

4. Compare and analyze your results

	CNN Supervised	CNN Self-training	Autoencoder as pre-trained CNN
training accuracy	~0.93	~0.92	0.3936
Best testing accuracy	0.6276	0.6674	0.3654

- 不意外的，self-training會比supervised再進步一些。
- Autoencoder可能因為我只拿來當CNN用，沒有做clustering，所以結果不會比Supervised好
- Autoencoder的test_acc/train_acc是三者之中最高的，而且這種高比值在前兩者的訓練過程中也從未出現，原因也可以想像：autoencoder在訓練時不會看image的label，所以抽出的feature會更具一般性，就更能避免overfitting發生。