HW2 MNIST

404410030 資工三 鄭光宇

環境設置：

使用 python，並使用 ipython notebook 環境。

使用 sklearn 套件，裡面有的 Cross-Validation 工具做交叉驗證、使用 t-SNE 對資料可視化。

使用 matplotlib 來繪製圖表。

使用 keras 來快速搭建 CNN 模型。

資料集：

使用這次作業指定的 MNIST。

MNIST：

28x28 大小的手寫數字，共有 10 種數字（0~9）。

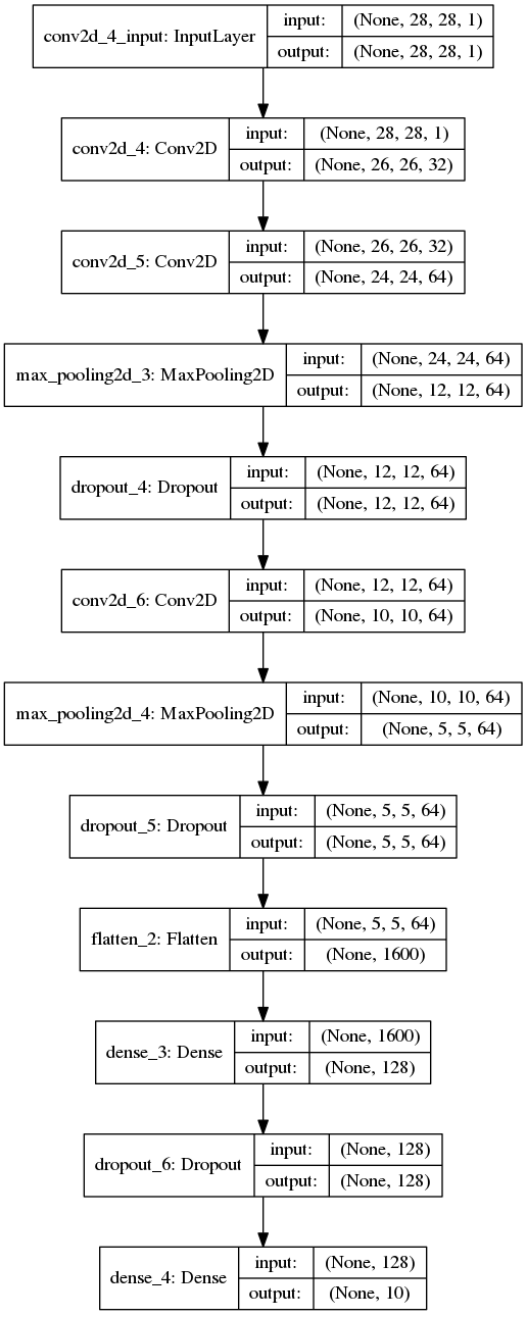
因為 keras 本身內建有 MNIST 資料集，所以沒有另外下載。

實驗結果：

MNIST：

我先將圖片數值 scale 到[0,1]的浮點數。

並使用如下 CNN 架構：



與原始 MNIST 上 CNN 的 LeNet-5 較大不同之處，是我有使用 Dropout，並多加了更多層 Convolution->Pooling，在這樣的架構下，training set 上的 5-fold Cross-Validation 可以達到 99.21±0.09% 左右，±0.09% 是 5-fold Cross-Validation 的兩個標準差範圍。

輸出的 activation function：softmax

選擇的 Loss function：categorical_crossentropy（多類別的對數損失）

Label Encoding：one-hot encoding

選擇的優化器：Adam (learning rate: 0.001)

訓練集上測試得 10-fold Cross-Validation 如下：

5-fold cross-validation
99.17%
99.10%
99.24%
99.37%
99.16%

Accuracy: 99.21±0.09%

之後，在測試集上面驗證效果。

測試集上的 Confusion Matrix 如下：

	0	1	2	3	4	5	6	7	8	9
0	978	0	0	0	0	0	1	0	0	1
1	0	1130	0	1	0	2	0	2	0	0
2	1	0	1021	0	0	0	0	10	0	0
3	0	0	0	1009	0	1	0	0	0	0
4	0	0	0	0	973	0	2	0	2	5
5	0	0	0	9	0	882	1	0	0	0
6	1	2	1	0	1	3	949	0	1	0
7	0	2	1	0	0	0	0	1024	0	1
8	0	0	2	2	0	0	0	0	969	1
9	1	0	0	2	3	1	0	1	1	1000

對於每一個類別，效能評估基準如下表：

CLASS	PRECISION	RECALL	F1-SCORE	SUPPORT
0	1.00	1.00	1.00	980
1	1.00	1.00	1.00	1135
2	1.00	0.99	0.99	1032
3	0.99	1.00	0.99	1010
4	1.00	0.99	0.99	982
5	0.99	0.99	0.99	892
6	1.00	0.99	0.99	958
7	0.99	1.00	0.99	1028
8	1.00	0.99	1.00	974
9	0.99	0.99	0.99	1009
AVG/TOTAL	0.99	0.99	0.99	10000

總結測試測試集的

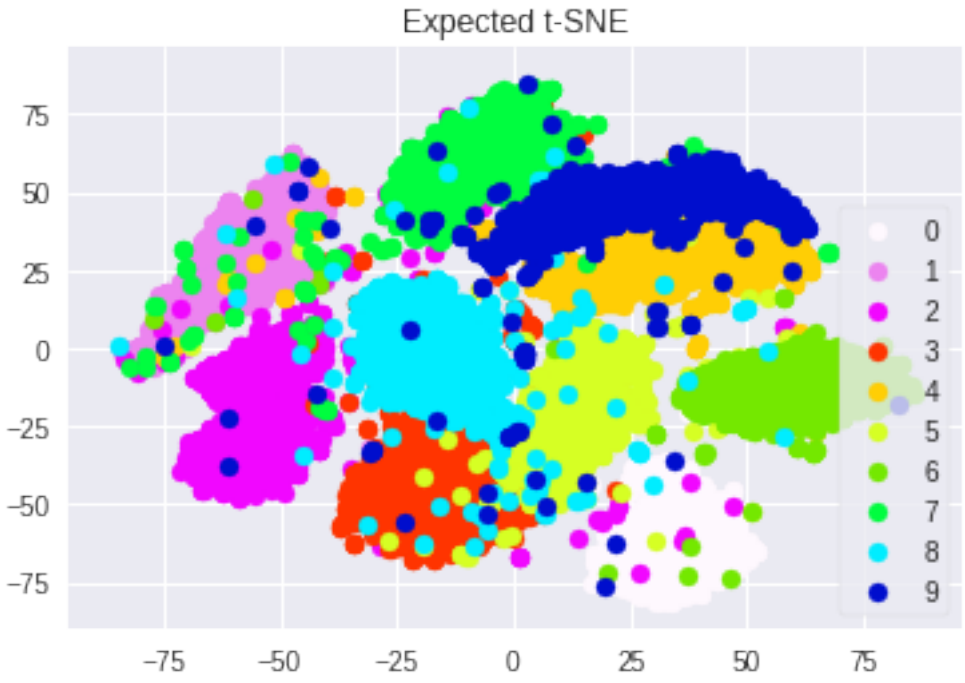
Error rate：0.65%

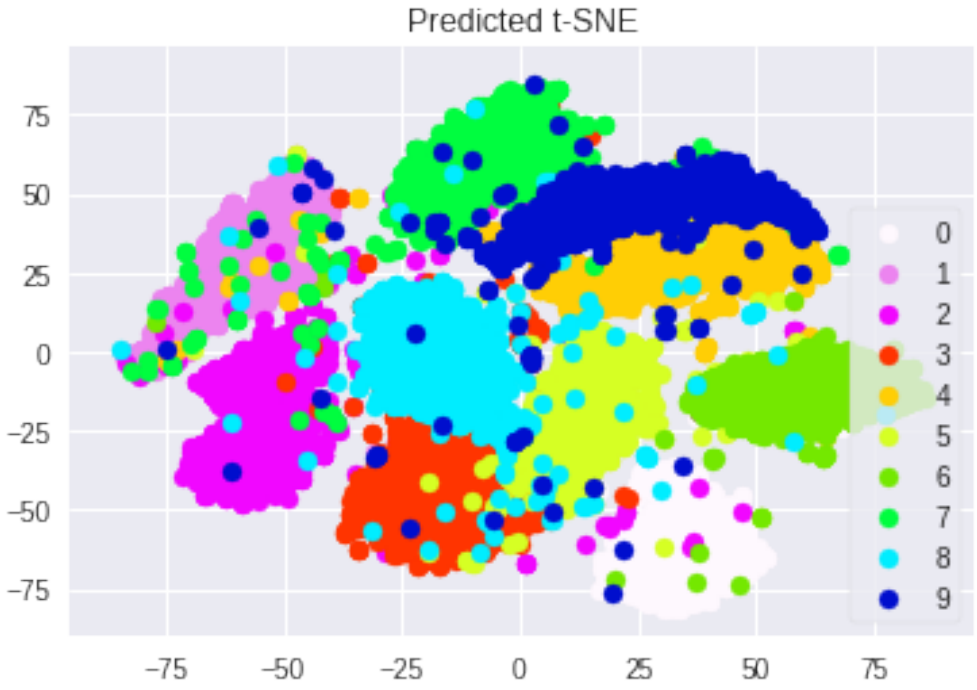
Accuracy：99.35%

可以看出效果非常好。

資料可視化：

使用最近流行的 t-SNE 將資料降維、投影到 2D 平面上，使資料可視化。測試集上可視化結果如下：





第二張圖片是 CNN 預測的結果，可以看出預測結果與 Ground Truth 很接近，CNN 很好地切開了各個類別。

與其他方法的比較：

與 MNIST 資料集網站上的「SVM, Gaussian Kernel」比較，

我們使用 CNN，可以得到 0.65% 的 error rate，而 SVM 是 1.4%，可以說是有顯著的進步。

與 MNIST 資料及網站上的其他 CNN 相比，我們實作的方法好過大部分 MNIST 資料集網站上的 CNN，得到很不錯的 0.65% 錯誤率。效果較好的原因可能是因為使用比較晚期才出現的 Dropout 技巧，使得模型較不容易發生 Overfitting。

結論：

這次使用完 CNN 後，了解到它是個優秀的模型，在 MNIST 上能夠輕鬆得到 99% 以上的正確率。

實作上遇過的問題：

雖然調整資料輸入的 batch size 有助於幫助平行化，有更好的 GPU 使用率，

不過過高的 batch size 有可能導致最後結果 accuracy 較低。

參考資料：

我的 github：

https://github.com/peter0749/Computer_Vision/tree/master/hw2_mnist_cnn

PCA+SVM MNIST github:

https://github.com/peter0749/Multimedia-Content-Analysis/tree/master/mnist

Keras Documentation:

https://keras.io/