手寫數字辨識的機器學習
(利用 sklearn 之 SVM 範例程式進行了解)

一開始，我選擇了從在機器學習領域算是入門的 MNIST 手寫數字辨識資料集進行探索，並在實際執行網路上的範例程式之前先了解 MNIST 的背景資料。
【Tensorflow Day3：熟悉 MNIST 手寫數字辨識資料集:
https://ithelp.ithome.com.tw/articles/10186473】

從我的理解，MNIST 是一套手寫數字的辨識資料集，其資料分為以下三種 da，是對於機器學習而言重要的分類:
1.training data :
　　用於機器學習的第一階段，
　　給予手寫數字的 image(X，通常是矩陣)、對應的 label(y，訓練資料 X 所對應的答案)。
2.test data (&)
3.validation data:
　　藉由 test data 和 validation data，驗證機器學習的效果。
　　(輸入 X 是否能夠對應到應該對應的 y 的比率)
【連結網址: http://yann.lecun.com/exdb/mnist/】

然而因為範例使用了難度較高的 tensorflow，因此決定以 sklearn 所提供的 Recognizing hand-written digits 進行實驗操作，因為兩者的目的其實相似，皆是以機器學習的方式辨識手寫數字，而在我目前的認知之中，它們最大的差別除了在使用的工具不同之外，還有圖片的像素不同，MNIST 每張圖片是 28*28=784 個像素，而 sklearn 的手寫數字辨識資料庫提供的是 8*8=64 像素的影像資料。(其他細節還有待更進一步的觀念釐清)

在更進一步了解這個範例程式之前，先回顧一下機器學習的本質。在進行機器學習的時候，我們需要給它明確的 feature(上述的 X)和明確的 label(上述的 y)，也就是在進行 training data 資料處理的時候，我們必須明確定義什麼樣的 feature 組合會對應到什麼樣的 label。每一個 feature 就是一個維度，不論過多或過少(稱為過擬合和欠擬合)，對於機器進行分類的訓練都是不佳的，因此，選定的 feature 種類與數目便很關鍵。
【參考資料:
https://hk.saowen.com/a/148d8364aba4184437da06429cbf2b1357a77995cbaddb98fc73a870895ae974】

在稍微複習機器學習的原理之後，就進一步去了解這個範例所使用的演算法。

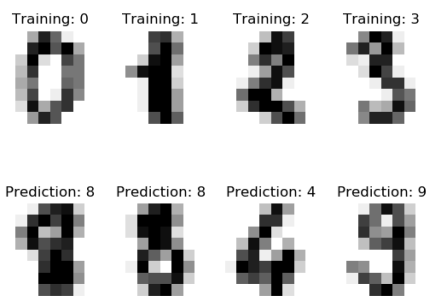
sklearn 所提供的 Recognizing hand-written digits 範例是利用課堂上所教分類演算法(Classification)中的支持向量機(Support Vector Machine，簡稱 SVM)進行分類，藉由找到一個超平面(hyperplane)去劃分兩個不同的集合，其優勢在於能夠處理小樣本之非線性及高維資料，而其應用則包含了此範例之手寫字體辨識、三維目標之辨識以及文本圖像的分類等實際情形。

在這次的練習操作之中，我所改變的是 svm.SVC 所吃的參數 gamma 的值。gamma 的物理意義在於它會影響每個支持向量對應的高斯作用範圍，也就是說，當 gamma 的值設得越大，其高斯分布越集中，作用範圍主要落在我們所提供的 training data 附近，對於 training data 的準確率高，但是對於未知的 test data 的分類效果便較差；如果將 gamma 的值設得越小，其效果則反之，對於 training data 的準確率較低，對於 test data 則不一定。因此，gamma 值不宜太大或太小。

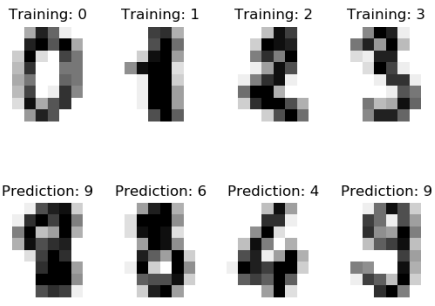
因此，為了驗證這個理論，我將 test data 和 training data 同時丟入模型中，並以手動的方式進行操作，求出 gamma 值在 0.01、0.001、0.0001、0.00001 時所得到的結果:

	0.01	0.001	0.0001	0.00001
precision - test data	0.92	0.97	0.94	0.83
precision - training data	1.00	1.00	0.98	0.87

從 training data 的結果中可以發現，當 gamma 值越小，將 training data 丟入模型所得之準確率越低，驗證了上述理論；而 test data 在四個嘗試中的最佳值 0.97 則落在 gamma 值為 0.001 的時候。然而在進行操作的時候不可能這麼麻煩，為了得到最佳模型而必須不斷手動調整 gamma 值，sklearn 提供了良好的服務，讓我們輸入有範圍的 gamma 值，能夠直接藉由計算幫助我們找到最好的模型。



test data / gamma=0.001



test data / gamma=0.00001

而這份範例程式的運作，是將一份資料切割成兩個部分，前半部分作為 training data，後半部分作為 test data，讓機器經過 training data 的訓練之後，以 test data 進行驗證。sklearn 所提供的範例程式碼，在訓練過程的邏輯部分提供了清楚的備註，雖然所使用到的函式會需要到網站上去查，但是在上完三天的 AI 課程，粗略了解一些些的機器學習基礎之後，如此複習方式的好處是比較能夠再進一步了解選定的其中一些機器學習模式，可以更深入去探討一個參數的影響。或許在實際操作的時候不一定會需要知道到那麼細，但是這樣是了解基礎很有趣的方式。

【程式碼參考自 sklearn 網站:http://scikit-learn.org/stable/auto_examples/classification/plot_digits_classification.html】