|  |
| --- |
| Zinian-期末專題 |
| 手寫數字辨識系統 |
|  |
| 以 MNIST 資料集加上Tensor Flow模組作為數據來源，透過DNN架構，訓練出一套文本資料，最後以Qt C++實現GUI視窗介面。 |
|  |
| **405410016楊淨、405410010毛胤年** |
| **2018/6/21** |
|  |



目錄  
1.背景   
2.動機   
3.目的   
4.系統開發 - Tensor Flow、Qt  
5.結果   
6.結論、心得   
7.參考文獻   
8.附錄 - 分工、進度表、通訊紀錄

1. 背景

在資訊科技日新月異的年代，科技扮演著進步的重要推手。其中人工智慧是未來產業升級的重要關鍵，無論是近期各類新創公司，還是傳統製造業，或多或少都有他們影子。而我們期初科技報告所談到的「語音辨識」，也有類似的架構，於是我們便想以此為主題，嘗試看看是否能了解運作的原理。

作為入門的學習者，雖然許多資料都能從網路上取得，但是礙於小組人數不多，我們選用較為容易實現的主題，數字手寫辨識。

1. 動機

對於PC端的程式來講，手寫輸入或許不比手機平板來得方便快速，畢竟鍵盤輸入對大眾來講已經很方便了。但如果有某天你的鍵盤剛好壞了、豆漿撒上去了、寵物嘔吐了、你家長輩不會打字等等，在一個特別的狀況時，有這麼一個功能或許會方便許多。

1. 目的

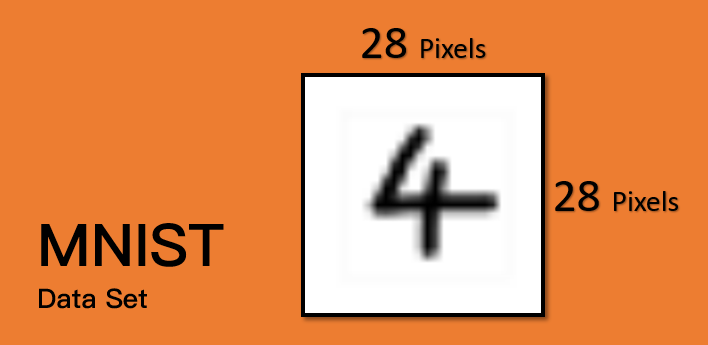
提供一個工作區，能夠實現手寫辨識，使用者只要將想要的數字繪製到畫面中，便能透過運算生成使用者想要的數字，並且直接複製到剪貼簿中，這樣便能直接使用貼上功能來輸入想要的數字。

1. 1系統開發 - Tensor Flow

Tensor Flow是目前最受歡迎的深度學習框架。開源的特性使得它得以借助社群共享的力量壯大，同時得益於這些社群的發展，網路出現大量Tensor Flow的學習資源，這也是身為初學者的我們選擇它的原因。在本次報告中我們會使用程式語言Python驅動 Tensor Flow，再搭配MNIST資料庫做為我們的訓練基礎。

DATA

首先MNIST資料庫中共有70,000張圖片，每一張圖片由784個像素組成；



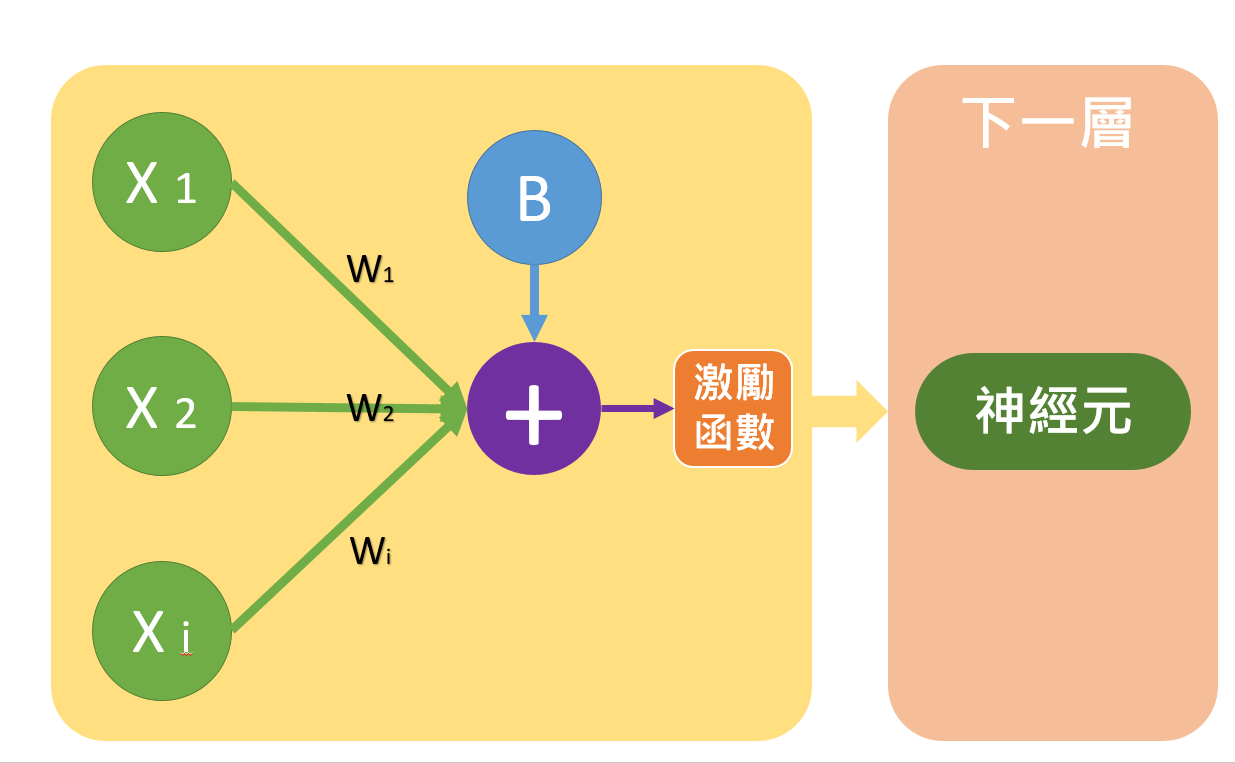
* 在機器學習的領域中，一般將70,000筆資料分成訓練資料55,000筆、驗證資料5,000筆與測試資料10,000筆。

DNN

在架構方面，我們使用**深度神經網路**DeepNeuralNetwork **(**DNN**)**來實現，它是一種模仿生物神經網路的數學模型，藉由抽取資料的特徵，統計並歸納問題**。**

**通常**DNN**的內部分為三層，分別是**Input Layer、Hidden Layer、Output Layer，每一層依據問題特性以及設計方針，都有各自的層數及神經元數量。其中Input Layer、Output Layer都只有一層，由問題的特性決定I/O的神經元數量，而Hidden Layer因為是負責抽取特徵值的重要關鍵，它的層數和神經元數需要由設計者自行決定，依據設計的好壞，直接影響模型的準確度。

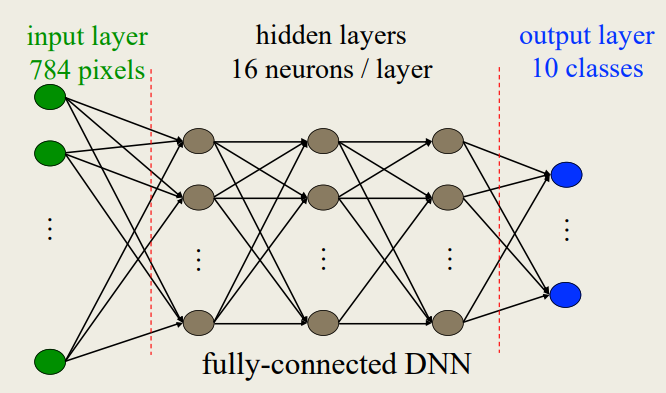
DNN 的各神經元皆代表數值，下圖可見第一層神經元Xi有對應的權重Wi(weight)，將這些Xi與對應權重相乘，再加上偏置B(bias)，並代入激勵函數可以得到下一層的神經元數值。



* 神經元若僅做加權計算，則輸入輸出脫離不了線性關係，喪失神經網路的意義，因此必須加入激勵函數
* 將上述套用在其他的神經元，我們可以依序得到 DNN 各神經元的數值

ModelTraining

我們先將28\*28 pixels的二維陣列拉成784 pixels的一維陣列，再將陣列輸入Input Layer。中間我們用了3層Hidden Layer(每層16個Node)來抽取圖片的特徵，經過數次的運算，最後Output Layer會將node收束到0~9，得出10組機率，取最高的當作Output。



4.2系統開發... Qt

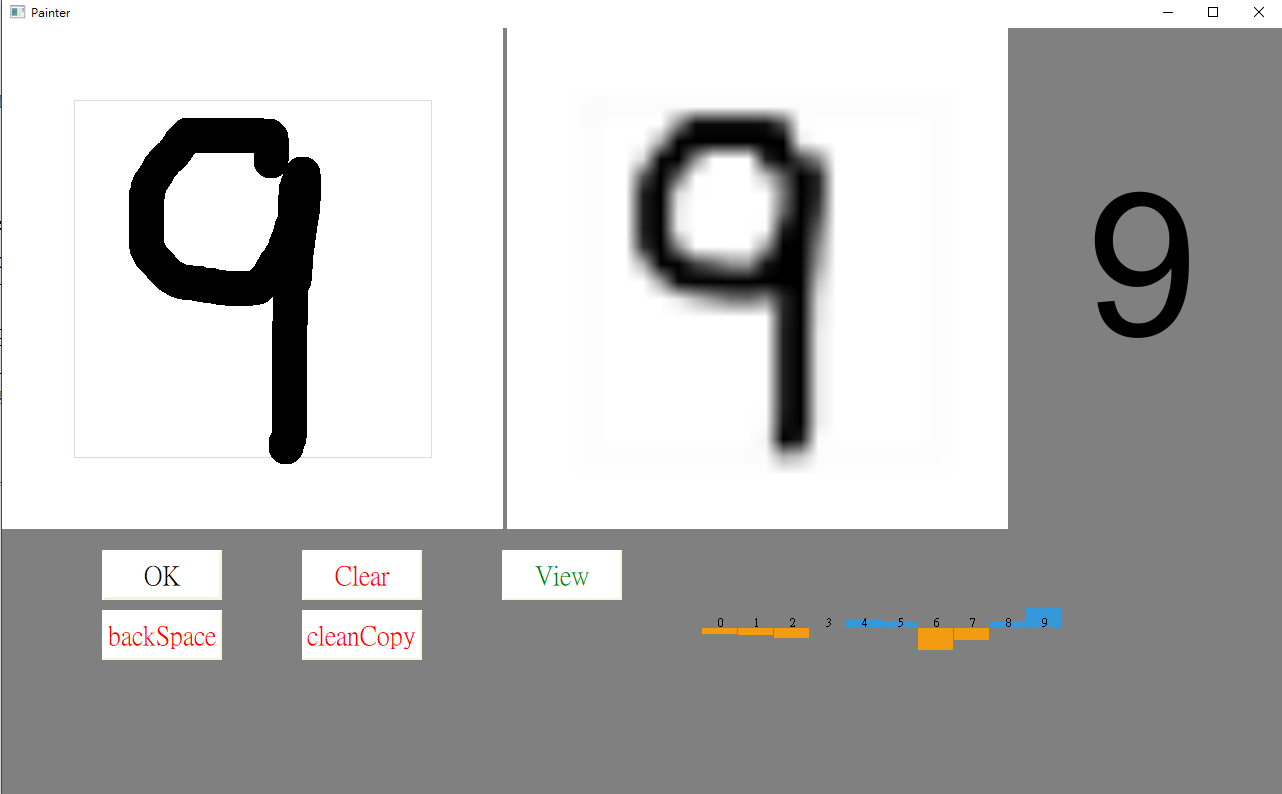
我們使用Qt框架開發Windows平台的手寫GUI，其中包含

|  |  |
| --- | --- |
| Input: | 手寫畫布、各式功能按鈕。 |
| Output: | 符合MNIST格式的圖片、各項機率分布、辨識結果。 |

考慮到之後要配合MNIST中的資料格式，在程式左方的手寫畫布上設有方框，提醒使用者要在方框內寫上數字，以利Qt產生符合MNIST格式的圖片，提高圖像辨識的準確度。

**使用方式:**

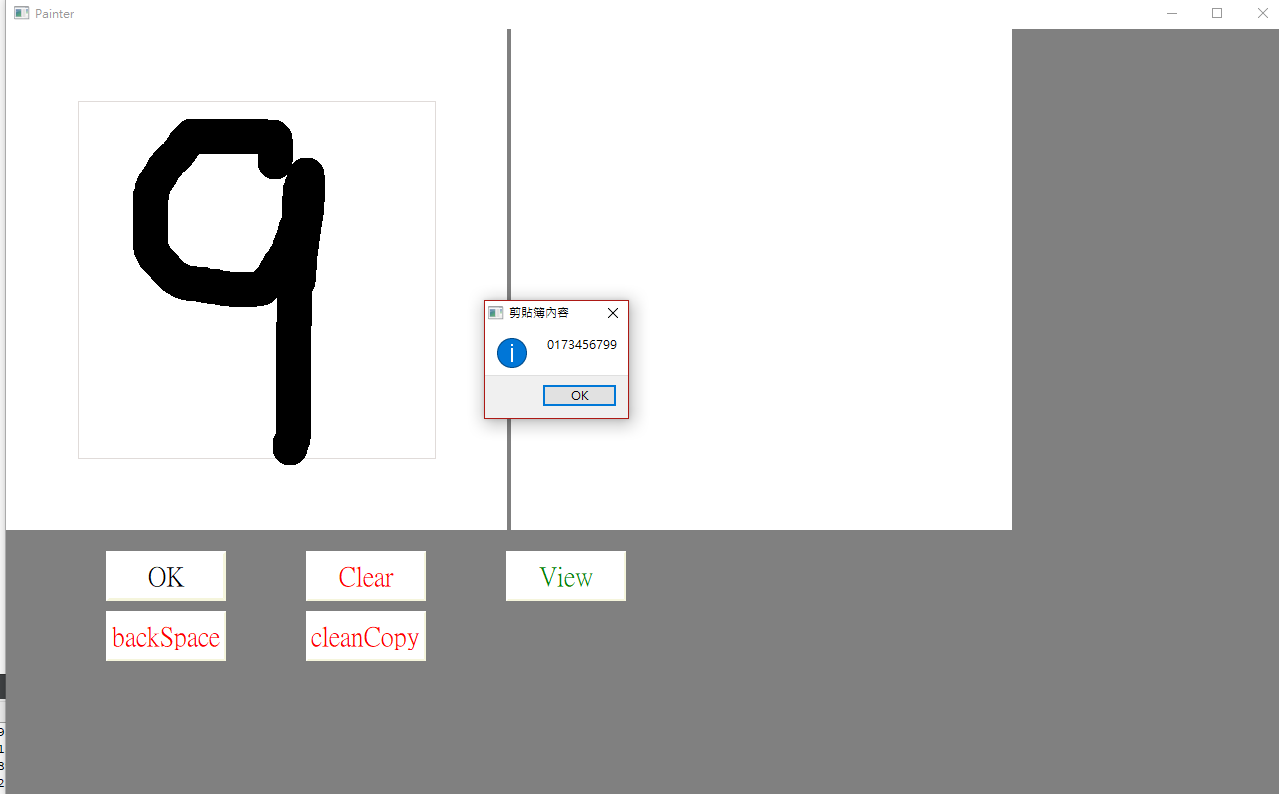
1. 在左上角的大繪圖區繪製數字。
2. 按下OK後，便會開始運算，並將計算結果文字附加到剪貼簿。
3. 按下View能觀察剪貼簿中的文字。
4. 按下clear能清除重繪。
5. 按下clean Copy能清除剪貼簿中內容。
6. 按下backspace 可以像鍵盤退格鍵一樣，將最後一個寫錯的字從剪貼簿中刪除。
7. 在想要的地方右鍵貼上吧!

****

1. 結果

多數的手寫數字都辨識成功，但是因為MNIST中的資料量仍然不足以應付變化莫測的手寫數字，因此當使用者寫出的數字與MNIST中的圖片差異太大時，仍然會有辨識錯誤的情況。

* 以下為連續手寫0-9之結果，辨識度不算低。



6.結論、心得

雖然結果有完成到最初的想法，但在開發過程中遇到了不少問題，到後來也會懷疑我們最一開始的企劃是否夠周全、程式的實用度、方便性等等，但由於我們的知識、能力、人手、時間上的各種限制，最終只得以做出了僅限於數字上的辨識系統，若能拓展到全中文、辨識度再高一點，或許能夠在實質意義上的實用，也能夠完全應付無鍵盤的狀況了。

美術風格我們自認還能加強，但在介面設計上，我們採取簡單快速的手段，一個輸入、一個結果能直接使用，你能連續寫出要用的數字，最後再用右鍵直接貼上就好。如果寫錯字、辨識錯誤，按下BACKSPACE也能刪除掉上一個字，要重新開始就用Clean copy來完成，若忘記到底剪貼簿中有什麼東西，按下view便能快速查看。

這次的作業讓我們學到了許多，比如說code的版本控制、工作時間的分配、工作內容的分配、如何將工作壓力平均分散等等，還有一些技術、環境上的問題，無論如何我們還是在預期的時間內將成品弄出來了。

7.參考文獻

QT教學:

<http://monkeycoding.com/>

**什麼是人工智慧、機器學習、深度學習？:** <https://blog.gcp.expert/ml-1-ai-ml-deep-learning-intro/>

tensorflow-教學庫: <http://darren1231.pixnet.net/blog/post/332753859-tensorflow%E6%95%99%E5%AD%B8----%E5%BB%BA%E7%BD%AE%E4%B8%80%E5%80%8Bcnn%E7%B6%B2%E8%B7%AF%E5%88%86%E8%BE%A8%E6%89%8B%E5%AF%AB%E8%BE%A8%E8%AD%98>

TechBridge技術共筆部落格: <https://blog.techbridge.cc/2018/01/27/tensorflow-mnist/>

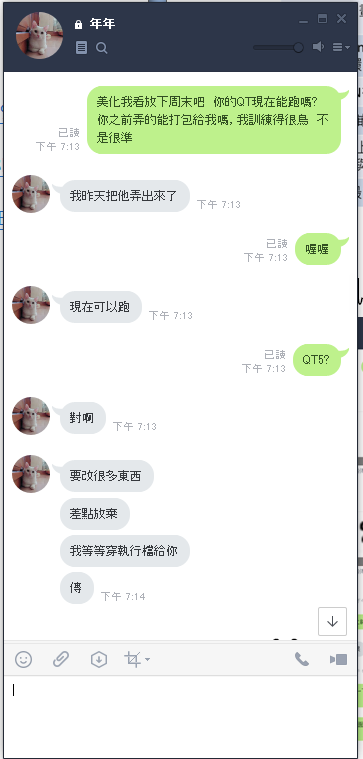
TensorFlow 实战:

<https://migege.com/post/tensorflow-practice-02-dnn-on-mnist>

1. 附錄(分工、進度表、通訊紀錄)





通訊紀錄: (僅列出部分)

