圖樣辨識

專題製作

期末報告書

|  |  |
| --- | --- |
| 主題: | 手寫數字辨識 |
| 組長: | 劉家豪 |
| 組員1: | 郭先旻 |
| 組員2: | 彭子豪 |

目錄

[壹、 前言 4](#_Toc515215911)

[一、動機 4](#_Toc515215912)

[二、目的 4](#_Toc515215913)

[貳、 研究方法 5](#_Toc515215914)

[一、 mnist數據集 5](#_Toc515215915)

[二、 softmax回歸模型 7](#_Toc515215916)

[三、 卷積神經網路(CNN) 10](#_Toc515215917)

[1.建立模型 11](#_Toc515215918)

[2.訓練模型 12](#_Toc515215919)

[3.測試模型 13](#_Toc515215920)

[參、 學習成果 14](#_Toc515215921)

[1. 隨機性 14](#_Toc515215922)

[2. Overfitting(過擬合) 14](#_Toc515215923)

[3. 如何建立出更好的模型 15](#_Toc515215924)

[肆、 參考資料 16](#_Toc515215925)

圖目錄

[圖 1 mnist數據解釋圖 6](#_Toc515215952)

[圖 2 softmax模型架構圖 7](#_Toc515215953)

[圖 3 softmax\_mnist 8](#_Toc515215954)

[圖 4 softmax\_mnist\_Accuracy 8](#_Toc515215955)

[圖 5 softmax\_mnist\_moreTraining 9](#_Toc515215956)

[圖 6 softmax\_mnist\_moreTraining\_Accuracy 9](#_Toc515215957)

[圖 7 CNN概念圖 10](#_Toc515215958)

[圖 8 CNN概念圖2 10](#_Toc515215959)

[圖 9 CNN\_mnist\_CreateModel 11](#_Toc515215960)

[圖 10 CNN\_mnist\_trainingModel 12](#_Toc515215961)

[圖 11 CNN\_mnist\_trainingData\_Accuracy 13](#_Toc515215962)

[圖 12 CNN\_mnist\_testData\_Accuracy 13](#_Toc515215963)

[圖 13 Overfitting 15](#_Toc515215964)

# 前言

## 一、動機

手寫數字辨識適合做為學習圖樣辨識、深度學習的入門題目，各個教學時常都以這道題目由淺入深的帶領學生學習

，此題目擁有著完善的資料、方法和數據，並且容易找到step by step的教學，我們這組所有人都是第一次接觸此領域，所以具有上述優勢的這項題目是很吸引我們的。

## 二、目的

學習是我們最大的目的，藉由理解手寫數字辨識的原理及運作，進而學習圖樣辨識的基礎，並在實作的過程中了解如何建立、訓練、測試模型，以及了解使用深度學習這項技術可能會遭遇的問題有哪些。

# 研究方法

## mnist數據集

最開始我們遇到的問題是訓練模型所需要的數據，一般的情況我們可能需要先自己拿起筆寫出無數個數字，接著把它存成圖像檔放進電腦，還要再將所有的圖像檔處理好並標記上正確的答案，最後還必須將一些錯誤的檔案給清理掉，避免模型學習到錯誤的知識，這些事情需要耗費我們大量的時間，但這並不是我們這項專題的重點，事實上除了我們也有非常多人有這同樣的問題，所以[Yann LeCun](http://yann.lecun.com/)、[Corinna Cortes](http://homepage.mac.com/corinnacortes/)、[Christopher J.C. Burges](http://research.microsoft.com/en-us/people/cburges/)三人提供了mnist數據集來解決這部分的問題。

mnist數據集中擁有大量且乾淨的手寫數字圖片數據，幫我們省去了蒐集數據、處理數據等等事項，讓我們可以把時間花在學習建立、訓練、測試模型上。

mnist數據來源： <http://yann.lecun.com/exdb/mnist/>

我們使用了mnist數據集中的60000筆training data，以及10000筆的test data。

每張圖片是 28 pixels\*28 pixels， 如下圖。

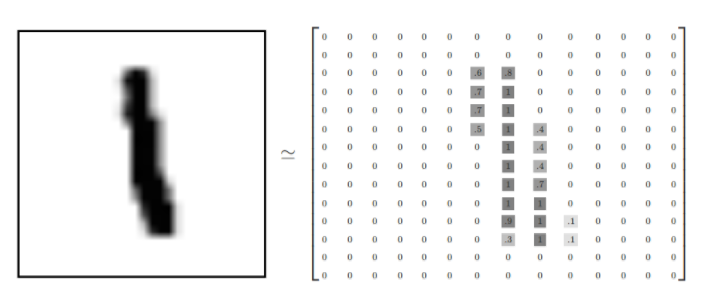


圖 1 mnist數據解釋圖

來源：https://fgc.stpi.narl.org.tw/activity/videoDetail/4b1141305d9cd231015d9d08fb62002d

## softmax回歸模型

SOFTMAX回歸模型：softmax 回歸模型是一個較為常見且基本的模型， sotfmax 模型會將輸入（input）的資訊，透過乘上**權重**（weight）與加上**偏差**（bias），再經由 softmax 函式轉換成機率的方式，得到每一種類別所代表的可能性並作為輸出（output），進而決定輸入的資訊是屬於哪一種類別。

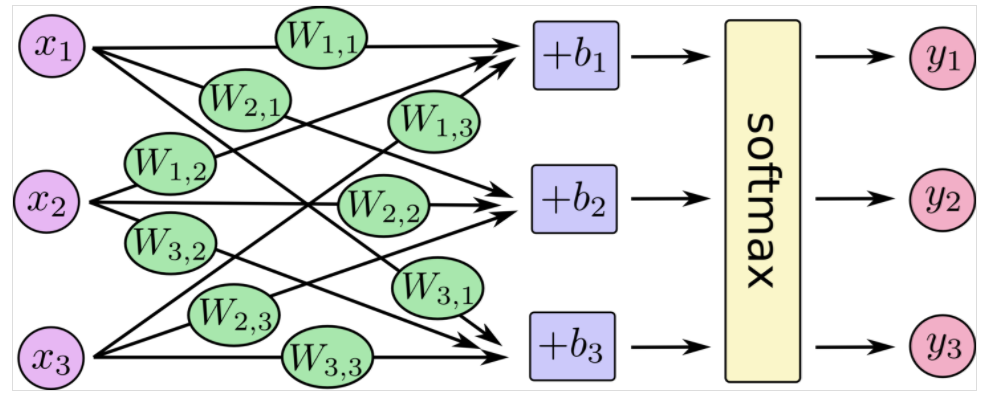


圖 2 softmax模型架構圖

來源:

https://fgc.stpi.narl.org.tw/activity/videoDetail/4b1141305d9cd231015d9d08fb62002d

最初我們跟著網路上的教學先建立出最簡單的softmax回歸模型來進行手寫數字辨識。

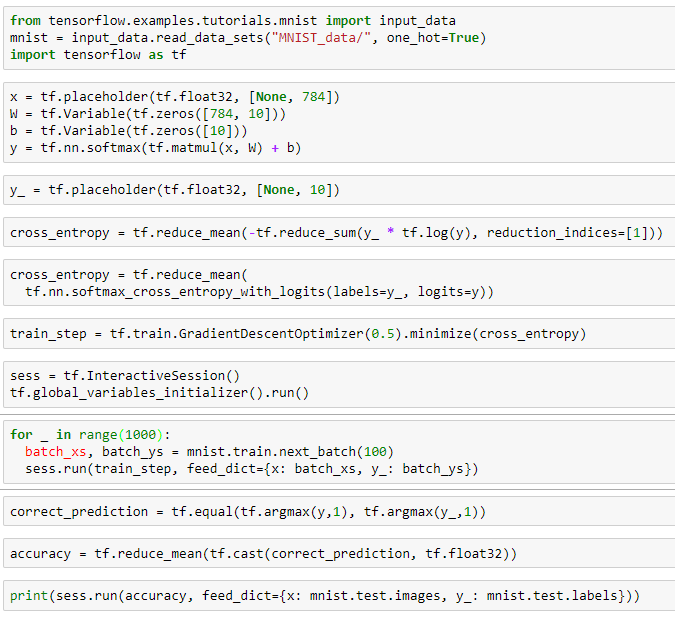


圖 3 softmax\_mnist

訓練完的模型測試結果為辨識成功率90.56%。

圖 4 softmax\_mnist\_Accuracy

接著我們大幅增加模型的訓練次數。



圖 5 softmax\_mnist\_moreTraining

訓練完的模型測試結果為辨識成功率為92.64%



圖 6 softmax\_mnist\_moreTraining\_Accuracy

經過多次測試，我們發現這個簡單的模型所能達到的極限大約落在辨識成功率92%，但離我們最初訂定的目標95%還有一段距離，所以接下來我們採用更進階的模型來達成更高的成功率。

教學資料來源：

<https://blog.gtwang.org/programming/tensorflow-softmax-regression-hand-written-digits-recognizer-tutorial/>

## 卷積神經網路(CNN)

卷積神經網絡(Convolutional Neural Network)簡稱CNN，CNN是所有深度學習課程、書籍必教的模型(Model)，CNN在影像識別方面的威力非常強大，許多影樣辨識的模型也都是以CNN的架構為基礎去做延伸。

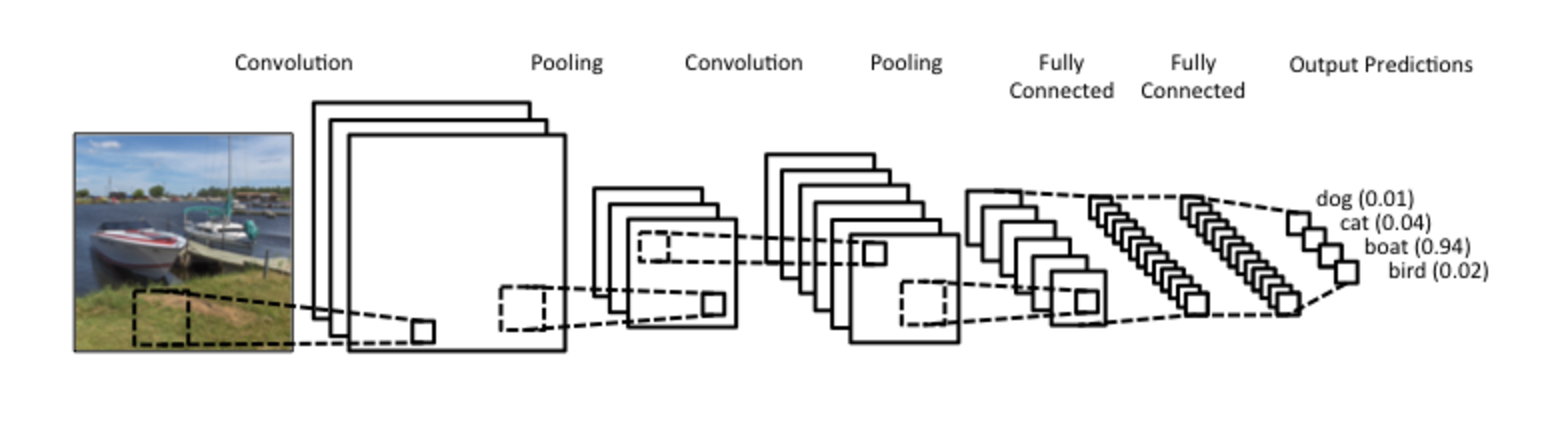


圖 7 CNN概念圖

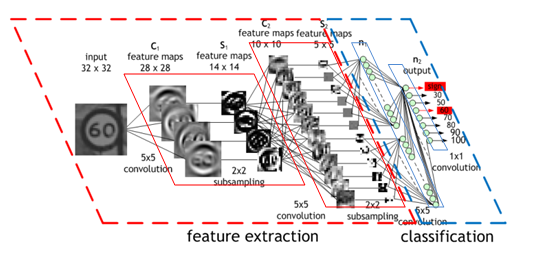


圖 8 CNN概念圖2

資料來源：參考資料[5]

### 1.建立模型

以上述的架構為原型，我們建立出了以下模型。



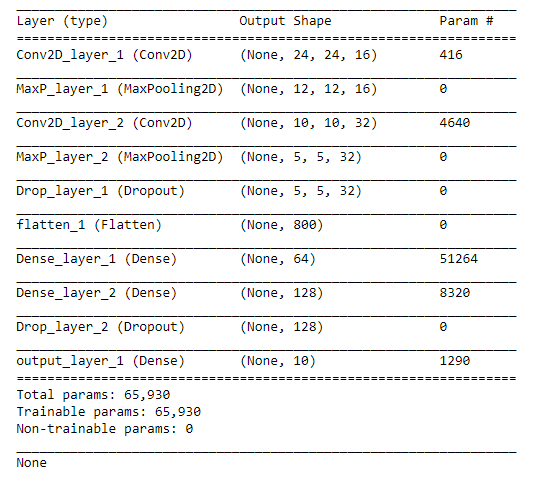


圖 9 CNN\_mnist\_CreateModel

### 2.訓練模型

接著使用training data來訓練此模型，訓練20次，大約花費了6分50秒。

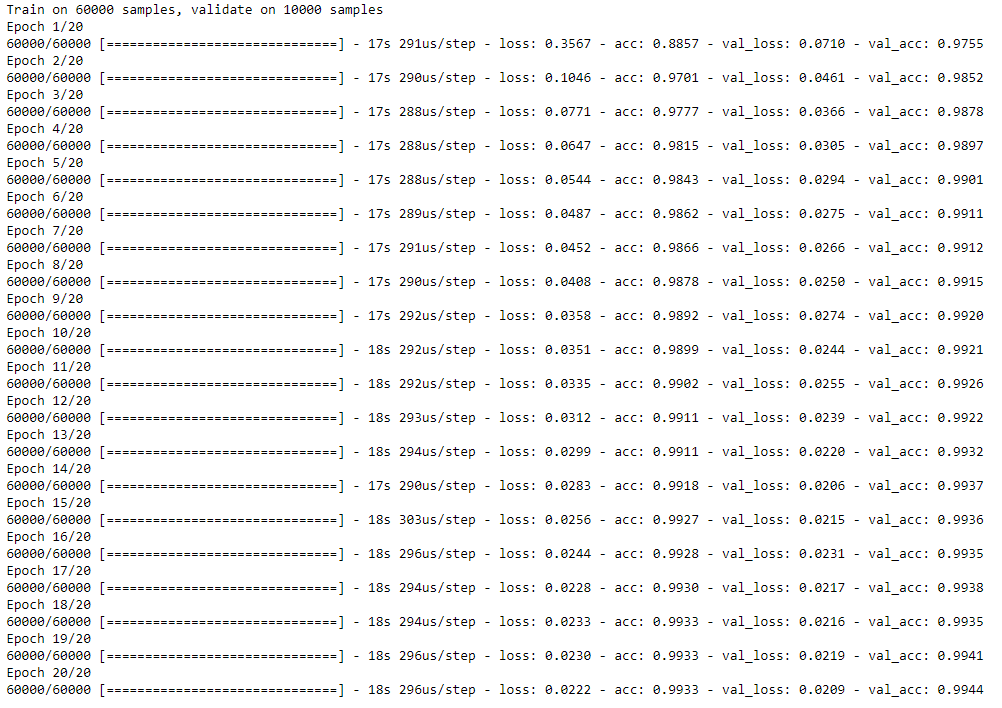


圖 10 CNN\_mnist\_trainingModel

### 3.測試模型

　　使用training data測試辨識成功率為99.8%

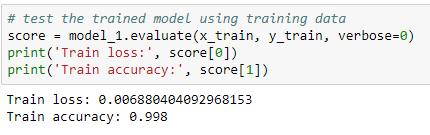


圖 11 CNN\_mnist\_trainingData\_Accuracy

　　使用test data測試辨識成功率為99.44%

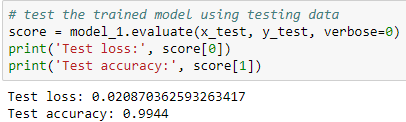


圖 12 CNN\_mnist\_testData\_Accuracy

# 學習成果

透過此專題我們已經了解如何建立、訓練、測試模型，並且了解到一些我們所遇到的問題。

### 隨機性

在訓練模型的時候，kernel是隨積的，被丟掉的神經元也是隨積的，初始的權重也是隨積的，同一個模型所訓練出來的結果也不相同，需要花費大量的時間來重新訓練才能得出較好的結果。

### Overfitting(過擬合)

以手寫數字辨識並且使用mnist數據集的情況來討論，發生Overfitting的機率比Underfitting(欠擬合)的機率高很多。

Overfitting會造成模型對於training data擁有非常高的辨識成功率，但對於test data的辨識成功率反而降低，我們建立的模型已經藉由使用Dropout來降低發生Overfitting的機率，但相對的也會增加Underfitting的機率。

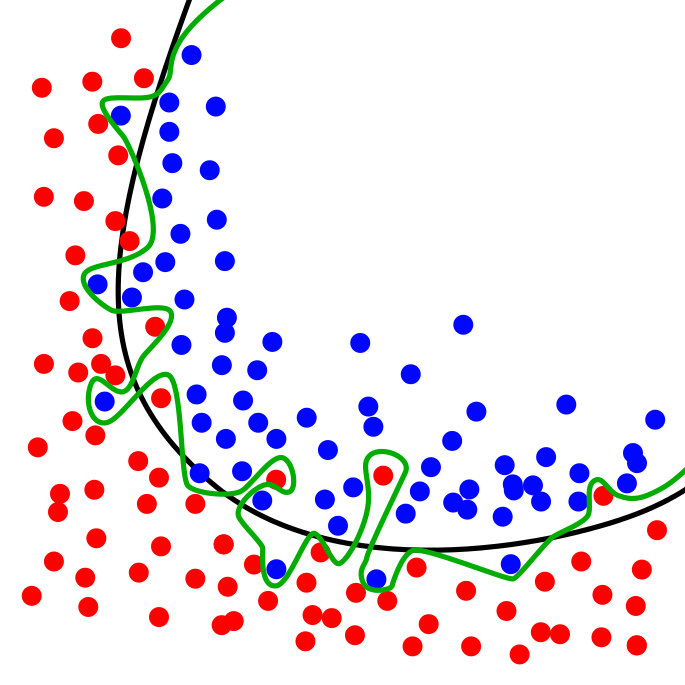


圖 13 Overfitting

圖片來源：<https://zh.wikipedia.org/wiki/%E9%81%8E%E9%81%A9>

### 如何建立出更好的模型

建立出更好的模型是一件很困難的事，根據理論較多層的深度學習模型會擁有更佳的效果，但由於隨機性和Overfitting都可能導致辨識成功率不增反減，建出更好的模型成為了一項大難題。

# 參考資料

[1]<http://yann.lecun.com/exdb/mnist/>

[2]<https://fgc.stpi.narl.org.tw/activity/videoDetail/4b1141305d9cd231015d9d08fb62002d>

[3]<https://fgc.stpi.narl.org.tw/activity/videoDetail/4b1141305d9cd231015d9d08fb62002d>

[4]<https://blog.gtwang.org/programming/tensorflow-softmax-regression-hand-written-digits-recognizer-tutorial/>

[5]<https://medium.com/@yehjames/%E8%B3%87%E6%96%99%E5%88%86%E6%9E%90-%E6%A9%9F%E5%99%A8%E5%AD%B8%E7%BF%92-%E7%AC%AC5-1%E8%AC%9B-%E5%8D%B7%E7%A9%8D%E7%A5%9E%E7%B6%93%E7%B6%B2%E7%B5%A1%E4%BB%8B%E7%B4%B9-convolutional-neural-network-4f8249d65d4f>

[6]<https://zh.wikipedia.org/wiki/%E9%81%8E%E9%81%A9>