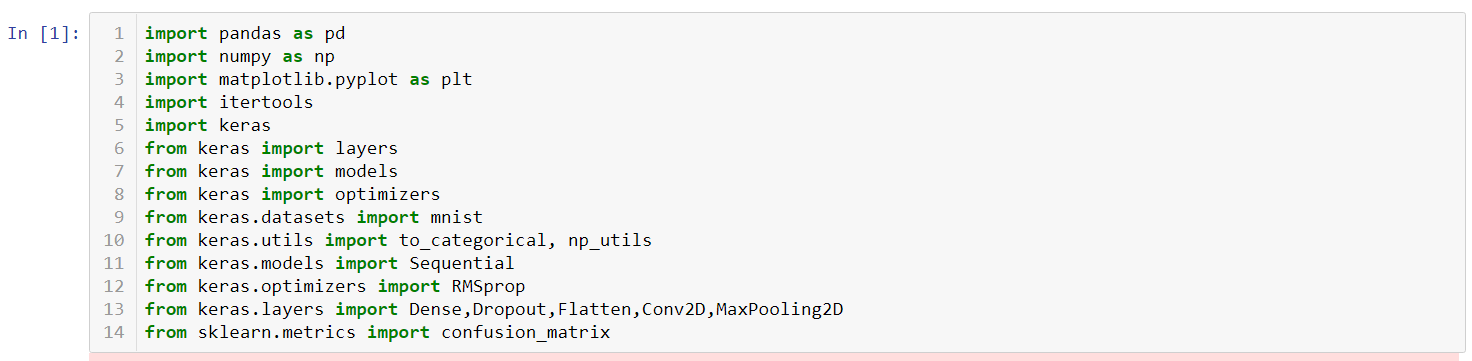
**MNIST：調整參數 (CNN準確率需高於MLP)**

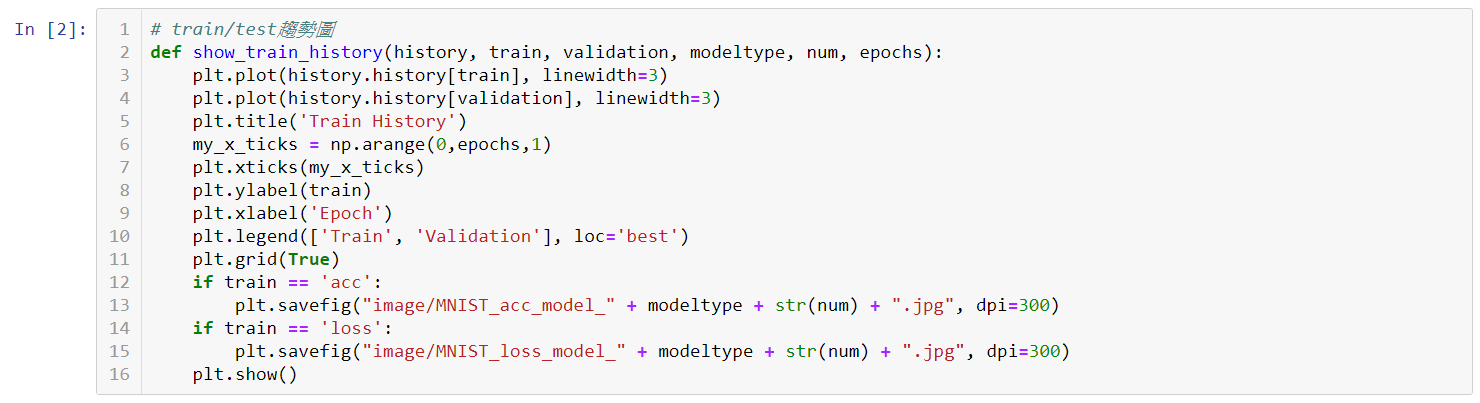
資管碩二 7107029022 邱靖詒

* **MNIST Handwritten Identification Dataset**

1. **Packages**

****

1. **Accuracy drawing function**

****

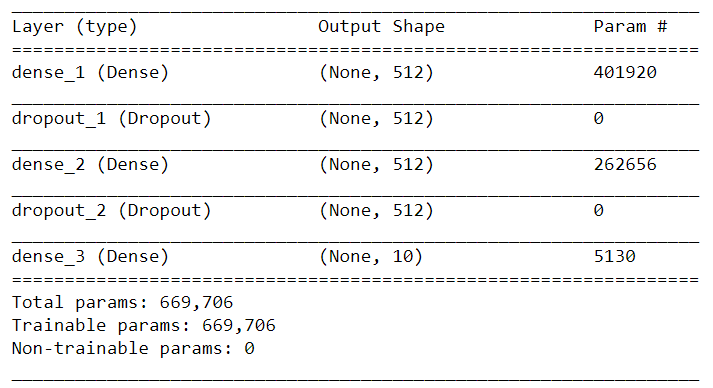
* **MLP**

1. **Definition of training set and testing set**



****

1. **Parameter setting**
   * + modeltype = 'MLP'
     + optimizer = 'rmsprop'
     + batch\_size = 128
     + num\_classes = 10
     + epochs = 20
     + verbose = 1
2. **Model summary**



1. **Result**

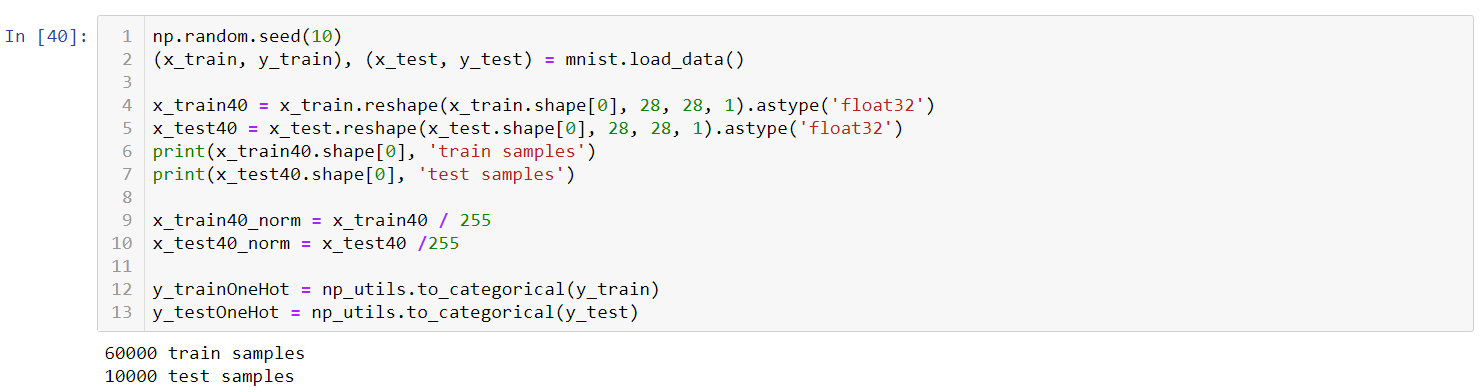
|  |  |
| --- | --- |
|  |  |
| Test Accuracy : 0.983200 | Test Loss : 0.112051 |

* **CNN**

1. **Definition of training set and testing set**

MLP因為直接送進神經元處理，所以60000筆轉換為一筆成28x28 = 784個神經元輸入。CNN因為必須先進行卷積和池化 (Max-Pool) 運算，所以必須保留影像的維度，因此60000 筆轉換成一筆成28 (長) x 28(寬) x 1(高)的影像單位。

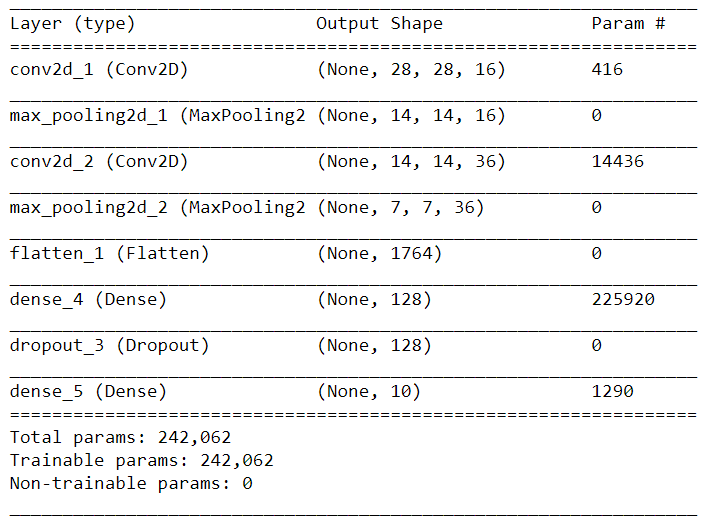
先把資料讀取與轉換，再把Features進行標準化與Label的One-Hot encoding。

****

1. **CNN Model 1**
2. **Parameter setting**
   * + - modeltype = 'CNN'
       - optimizer = 'sgd'
       - batch\_size = 64
       - epochs = 20
       - verbose = 1
3. **Model 1 summary**

模型設計Max-Pooling運算可以把影像縮減取樣(downsampling)，比如原本影像是4x4，經過Max-Pooling運算後，影像大小為2x2，其優點為減少需要處理的資料點、讓影像位置的差異變小、參數的數量和計算量下降(避免Overfitting狀況)

padding：補0策略，使用「same」，代表保留邊界處的卷積結果，通常會導致輸出shape與輸入shape相同。

****

卷積層1 + 池化層1

卷積層2 + 池化層1

神經網路(平坦層、隱藏層、輸出層)

1. **Result**

|  |  |
| --- | --- |
|  |  |
| Test Accuracy : 0.988400 | Test Loss : 0.035157 |

1. **CNN Model 2**
2. **Parameter setting**

* modeltype = 'CNN'
  + - * optimizer = 'rmsprop'
      * batch\_size = 128
      * epochs = 20
      * verbose = 1

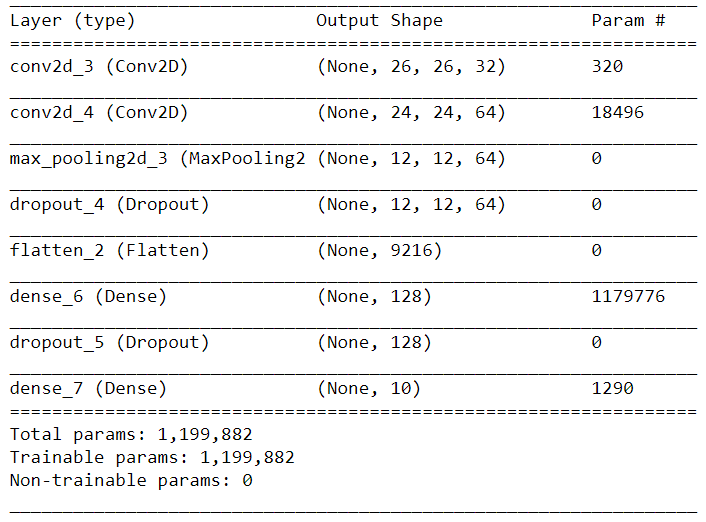
與CNN Model 1相比，修改優化器為RMSprop，batch size改為128。

1. **Model 2 summery**

與CNN Model 1相比，

相同：激發函數RELU

不同：修改kernel\_size為3x3，filters輸出維度也有調整，Dropout有0.25和0.5。

****

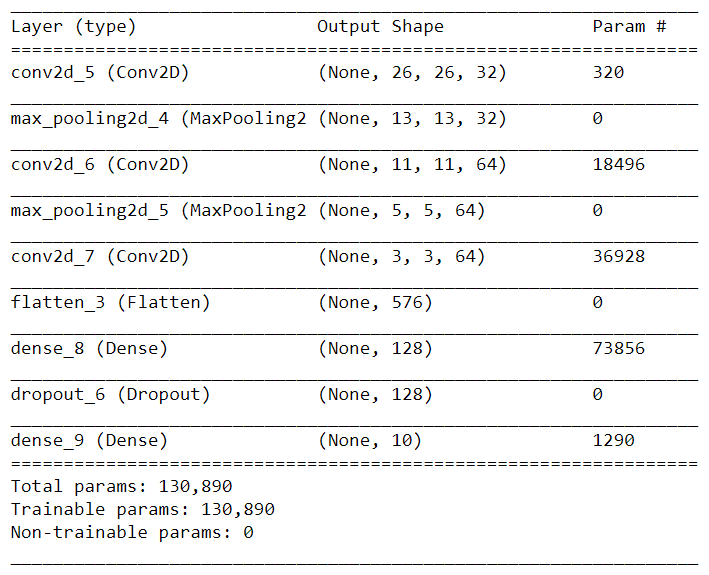
8. 使用32個卷積濾波器，每個濾波器的大小為3x3。
9. 使用64個卷積濾波器，每個濾波器的大小為3x3。
10. 選擇最佳者進行池化。
11. 隨機打開和關閉神經元來改善收斂。
12. 因維度大而使用平坦，只需要輸出分類。
13. 全連接層以獲取所有相關數據。
14. 使數據更加收斂。
15. 輸出softmax把矩陣壓縮為輸出機率。
16. **Result**

效果比CNN Model 1好一些。

|  |  |
| --- | --- |
|  |  |
| Test Accuracy : 0.989800 | Test Loss : 0.037317 |

1. **CNN Model 3**
2. **Parameter setting**
   * + - modeltype = 'CNN'
       - optimizer = 'rmsprop'
       - batch\_size = 128
       - epochs = 20
       - verbose = 1
3. **Model 3 summery**

與CNN Model 2相比，修改Dense層之參數。

****

1. **Result**

CNN Model 1、CNN Model 2、CNN Model 3三個模型，以CNN Model 3效果最好，準確率高達0.997。

|  |  |
| --- | --- |
|  |  |
| Test Accuracy : 0.997400 | Test Loss : 0.0.12839 |

* **CNN Model Selection Experiment**

CNN架構有很多選擇，如何選擇「最佳」的模型架構？「最佳」之定義可以是架構最簡單的，也可以是架構能有效的提高準確率，以下是針對MNIST手寫數字辨識資料集來提出不同CNN架構實驗。

代號：

24C5 代表使用filter的kernel size是5x5和stride為1的卷積層，帶有24 feature maps 。

24C5S2 代表使用filter的kernel size是5x5和stride為2的卷積層，帶有24 feature maps 。

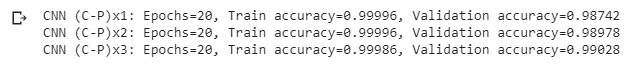
P2 代表使用filter的kernel size是2x2和stride為2的最大池化層。

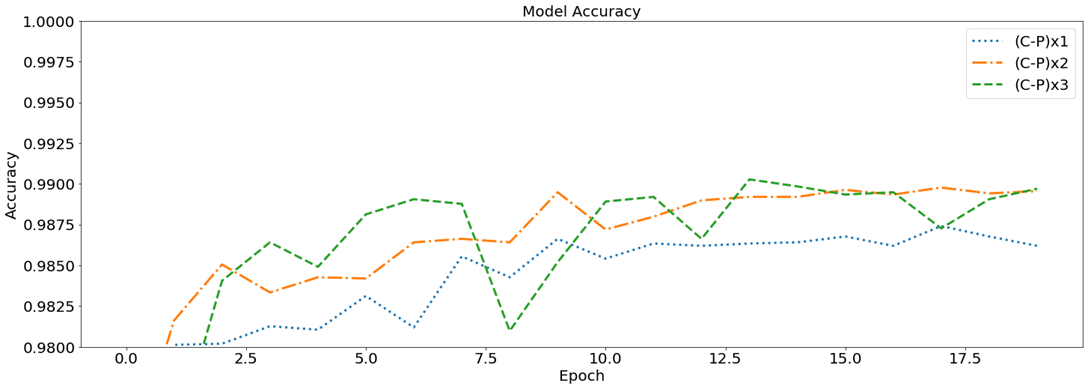
256 代表256個單元的全連接層。

1. **Convolution-subsampling Pairs**

input image 28x28 🡪 one pair 14x14 🡪 two pairs 7x7🡪 three pairs 4x4

結果：





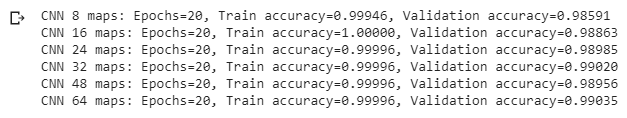
從上面的實驗中，three pairs卷積效果看似比two pairs卷積還要好，但為了提高效率，這種改善方法並不能保證不會產生額外的計算成本，因此選擇使用two pais。

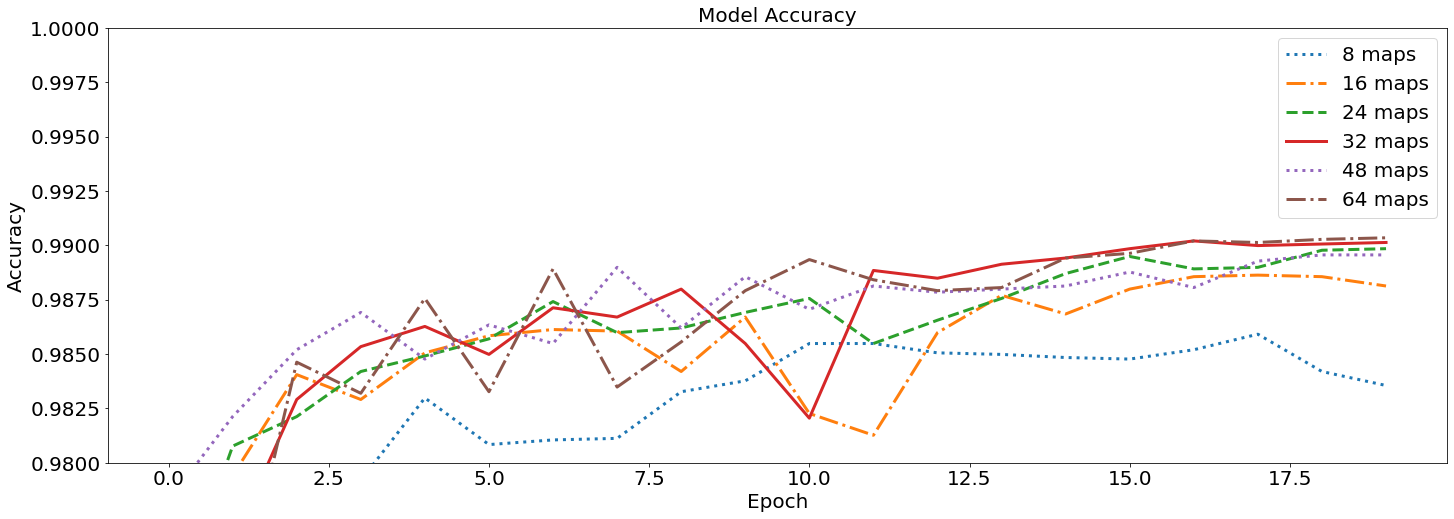
1. **Feature Maps**

根據上一個步驟，認為two pairs已足夠，現在要決定要用多少feature maps。

* [8C5-P2] - [16C5-P2]
* [16C5-P2] - [32C5-P2]
* [24C5-P2] - [48C5-P2]
* [32C5-P2] - [64C5-P2]
* [48C5-P2] - [96C5-P2]
* [64C5-P2] - [128C5-P2]

結果：





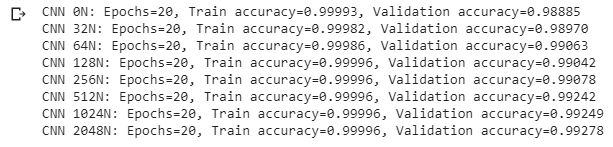
從上面的實驗可以看出，第一卷積層中的32個maps和第二卷積層中的64個maps是最好的。

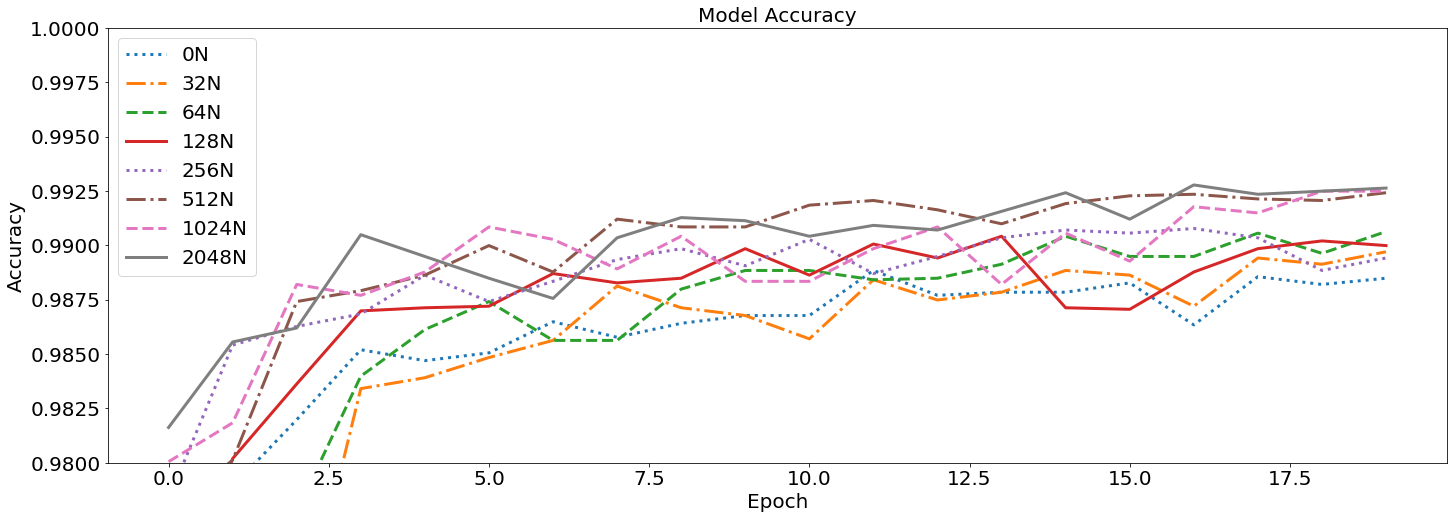
1. **Dense Layer**

根據上一個步驟，決定使用32和64個feature maps，現在要決定要用多少層dense layer。

* [32C5-P2] - [64C5-P2] - 0
* [32C5-P2] - [64C5-P2] - 32
* [32C5-P2] - [64C5-P2] - 64
* [32C5-P2] - [64C5-P2] – 128
* [32C5-P2] - [64C5-P2] - 256
* [32C5-P2] - [64C5-P2] – 512
* [32C5-P2] - [64C5-P2] - 1024
* [32C5-P2] - [64C5-P2] - 2048

結果：



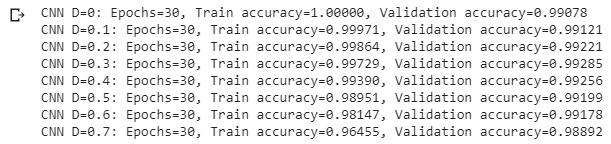


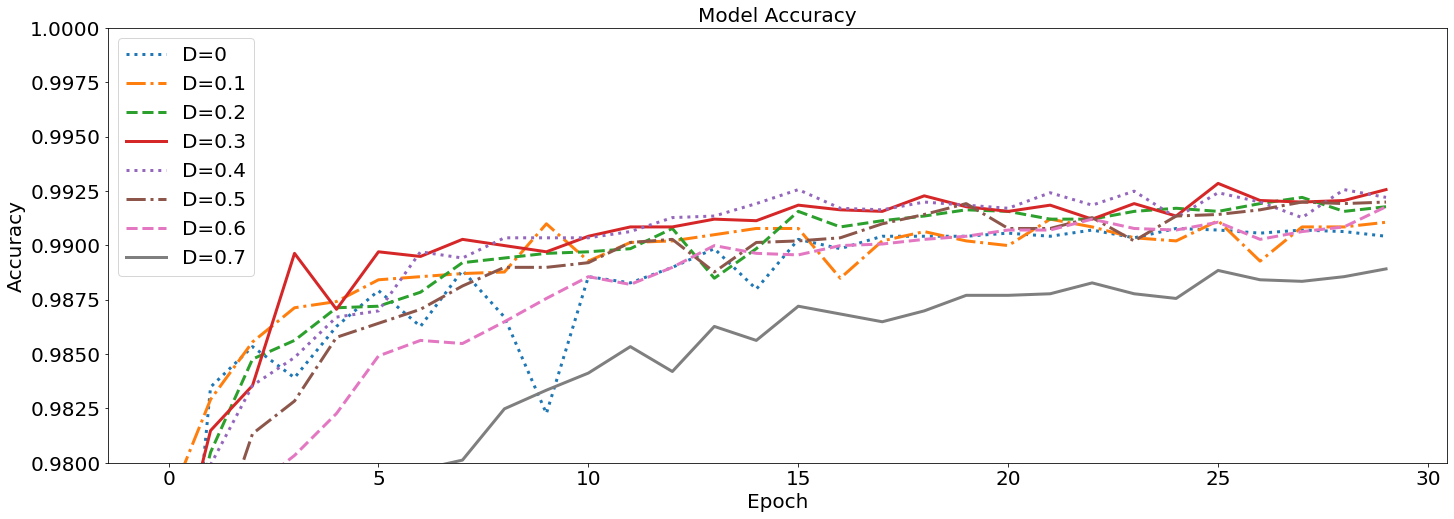
從實驗結果得知，以Dense的units設為128 (輸出維度為128) 效果最好。

1. **Dropout**

Dropout可防止過擬合，分為0%、10%、20%、30%、40%、50%、60%、70%八種情況。

結果：



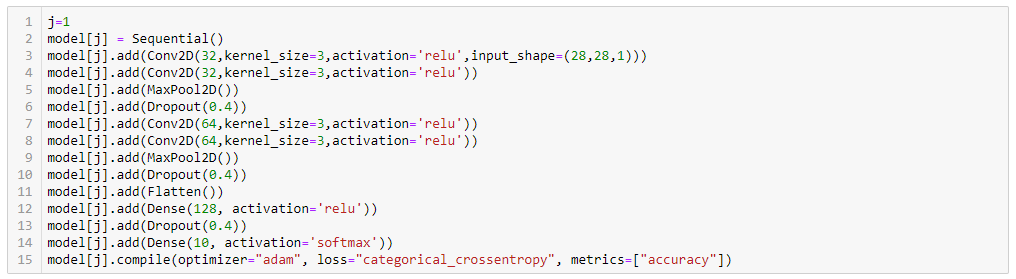


從實驗中發現，以Dropout為0.3的效果最好。

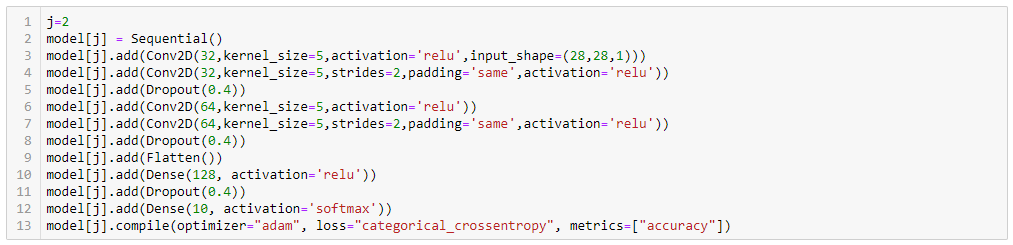
1. **Advanced Features**

除了使kernel size為5x5以外，也可使用兩個連續的3x3，並搭配使用strides = 2的捲積層進行二次採樣，而不是使用最大池化層，最後再加上批量標準化batch normalization和資料擴增data augmentation。

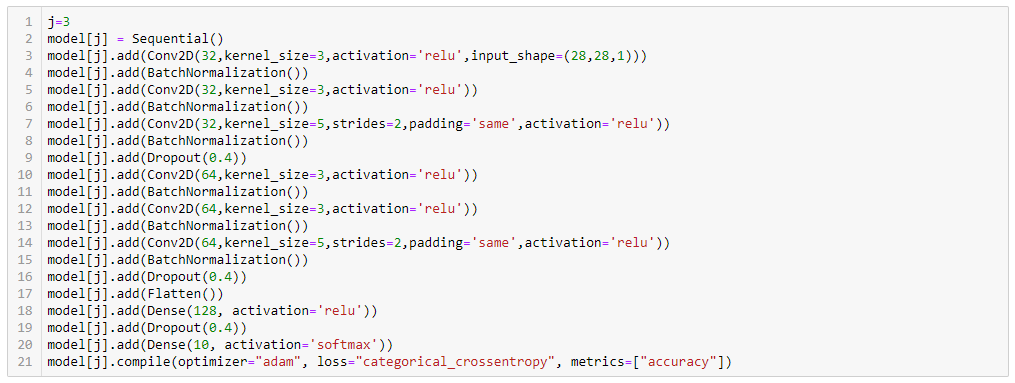
* 使用 '32C3-32C3' 來取代 '32C5'



* 使用'32C5S2' 來取代 'P2'

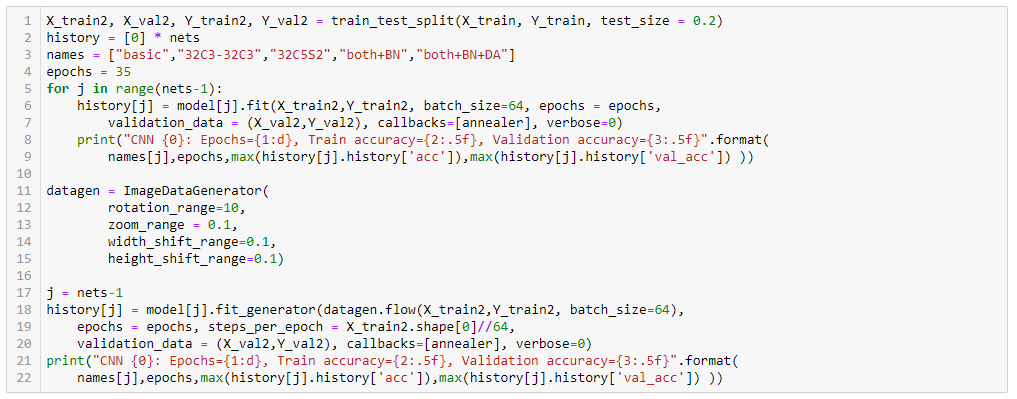


* 增加batch normalization

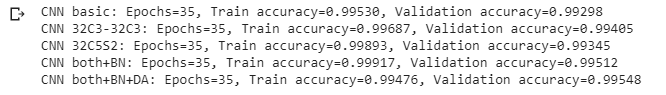


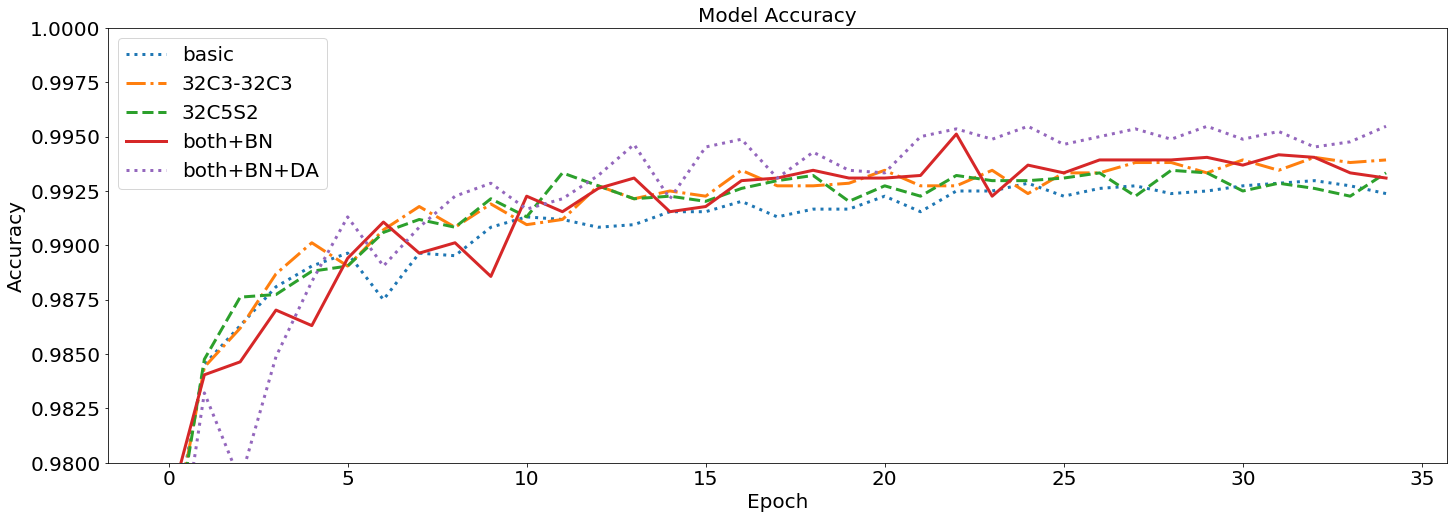
* 增加data augmentation





結果：





實驗結果發現，四種方法都提高了模型的準確率，而最後一個步驟將四種方法結合在一起，其準確率高達99.5%，或許訓練時間迭代更多次，效果會更好。

1. **Conclusion**

綜合以上實驗的結果，MNIST手寫數字辨識資料集的「最佳」模型為：

1. 使用[32C3-32C3-32C5S2] - [64C3-64C3-64C5S2] - 128的架構。
2. 搭配Dropout為0.3 (=30%)、增加batch normalization、增加data augmentation。