**程式說明:**

Add.npy => 加法練習

Sub.npy => 減法練習

Mix.npy => 加減混合

AdderandSubtractorPractice.py => 前處理

Model.npy => 模型訓練

**心得:**

似乎只用一般的dense沒有辦法有很高的accuracy，似乎需要LSTM或RNN效能才會比較好，即便是train dataset的accuracy也會卡在0.5左右上不去，因該是因為需要知道以前的數字才能好的預測結果。LSTM hidden layer 256 則會出現over fitten，Accuracy on train會到0.9但Accuracy on test依然很低。把hidden layer調成128後發現accuracy依然沒有變高。

後來開始調整訓練資料數量後發現Accuracy on test開始上升，但加道

資料量1000000時，反而又下降了。

**最後發現是模型不能夠在最後一層才reshap成4\*13，需要再訓練的時候就是4維一起訓練。**

**加法與減法結果:**

|  |  |
| --- | --- |
| add:    Accuracy on test : 0.9821 | sub:    Accuracy on test : 0.9750 |

**混合模型結果與比較:**

一 結果

二 比較不同模型:

Train 80000 test 20000

|  |  |
| --- | --- |
| NN dense 64:    Accuracy on test : 0.2971 | NN dense 256:    Accuracy on test : 0.345 |
| LSTM model 256 :    Accuracy on test : 0.3642 | LSTM with RepeatVector first :  Accuracy on test : 0.3117 |

三 比較不同訓練筆數:

Model為LSTM model 128

|  |  |
| --- | --- |
| Train 20000 | Accuracy on test : 0.0042 (會越訓練越小) |
| Train 200000 | Accuracy on test : 0.2299 |

五 比較不同 batchsize

|  |  |
| --- | --- |
| 1000 : | Accuracy on test : 0.154 (會越訓練越小) |
| 100 : | Accuracy on test : 0.2299 |