model設計思路：

一開始我先用一層lstm和4個dense，每個dense的input都是lstm的output，結果準確率理所當然的只有84%，接著我就開始思考該怎麼改進。

首先是因為lstm因為forget rate的關係，會比較不在意前面的資料，但是加減法的前後數字都很重要

所以我使用了bidirection來讓lstm變成雙向的，如此一來前面數字的重要性就會提升

而實驗結果顯示準確率變成91%

再來是後面的dense，我猜想人類在算加減法時，通常都是一個位數一個位數的算，因此我在dense跟第一個lstm中間插入了另一個lstm，input則是上一個雙向lstm的結果重複4次，因此這個lstm會做4次，並每做一次就會吐出一個output，並分別接上4個dense。以此模擬人類一位一位算的方式

而實驗結果，準確率來到94%

最後一個則是想到，因為比較高位的數字會受到低位數字的影響，比方說百位數會因為十位數進位而有所變化，因此第二個lstm的結果應該要是從低位往高位才對。

而實驗結果準確率直接登上了99%，可見這個因素影響很大

剩下的就只是根據training 結果開始調整lstm的hidden units個數想，來小幅提升準確率而已。

實驗數據：

3 bit digit

Add

Training set : 18000

Valid set : 2000

Testing set : 60000

|  |  |  |  |
| --- | --- | --- | --- |
| The number of training epochs | Train accuracy | Valid accuracy | Testing accuracy |
| 10 | 0.9001 | 0.9229 | 0.7362 |
| 20 | 0.9812 | 0.9910 | 0.9605 |
| 50 | 1.0000 | 0.9984 | 0.9916 |
| 75 | 1.0000 | 0.9986 | 0.9930 |
| 100 | 1.0000 | 0.9984 | 0.9929 |

3 bit digit

Sub

Training set : 18000

Valid set : 2000

Testing set : 60000

|  |  |  |  |
| --- | --- | --- | --- |
| The number of training epochs | Train accuracy | Valid accuracy | Testing accuracy |
| 10 | 0.9587 | 0.9562 | 0.8567 |
| 20 | 0.9708 | 0.9850 | 0.9501 |
| 50 | 1.0000 | 0.9994 | 0.9991 |
| 75 | 1.0000 | 0.9999 | 0.9996 |
| 100 | 1.0000 | 0.9998 | 0.9997 |

3 bit digit

Add & Sub

Training set : 18000

Valid set : 2000

Testing set : 60000

|  |  |  |  |
| --- | --- | --- | --- |
| The number of training epochs | Train accuracy | Valid accuracy | Testing accuracy |
| 10 | 0.7895 | 0.7067 | 0.1094 |
| 20 | 0.9823 | 0.9614 | 0.8566 |
| 50 | 0.9988 | 0.9789 | 0.9253 |
| 75 | 1.0000 | 0.9947 | 0.9819 |
| 100 | 1.0000 | 0.9946 | 0.9831 |

分析：

加法和減法的準確度都是99%，但加減混和的可能是因為規則較為複雜，所以準確率稍微低了一點。

減法的準確率比加法高，有可能是因為減法的需要預測的位數比加法少，所以也比較容易預測，畢竟加法的數字會越加越多(會進位)，而減法會越減越少。

3 bit digit

Add & Sub

Training set : 36000

Valid set : 4000

Testing set : 40000

|  |  |  |  |
| --- | --- | --- | --- |
| The number of training epochs | Train accuracy | Valid accuracy | Testing accuracy |
| 10 | 0.9737 | 0.9517 | 0.8469 |
| 20 | 0.9940 | 0.9956 | 0.9842 |
| 50 | 0.9987 | 0.9988 | 0.9966 |
| 75 | 1.0000 | 0.9999 | 0.9987 |
| 100 | 1.0000 | 0.9999 | 0.9990 |

3 bit digit

Add & Sub

Training set : 54000

Valid set : 6000

Testing set : 20000

|  |  |  |  |
| --- | --- | --- | --- |
| The number of training epochs | Train accuracy | Valid accuracy | Testing accuracy |
| 10 | 0.9945 | 0.9969 | 0.9871 |
| 20 | 0.9999 | 0.9998 | 0.9988 |
| 50 | 1.0000 | 0.9999 | 0.9994 |
| 75 | 0.9999 | 0.9992 | 0.9968 |
| 100 | 1.0000 | 1.0000 | 0.9997 |

分析：

Training data越多，就越容易抓到規則，準確率也會越高。

2 bit digit

Add & Sub

Training set : 180

Valid set : 20

Testing set : 600

|  |  |  |  |
| --- | --- | --- | --- |
| The number of training epochs | Train accuracy | Valid accuracy | Testing accuracy |
| 10 | 0.4278 | 0.4000 | 0.0117 |
| 20 | 0.4500 | 0.5000 | 0.0100 |
| 50 | 0.5981 | 0.4833 | 0.0500 |
| 75 | 0.7574 | 0.5000 | 0.0400 |
| 100 | 0.9315 | 0.4667 | 0.0317 |

2 bit digit

Add & Sub

Training set : 1800

Valid set : 200

Testing set : 6000

|  |  |  |  |
| --- | --- | --- | --- |
| The number of training epochs | Train accuracy | Valid accuracy | Testing accuracy |
| 10 | 0.4511 | 0.4500 | 0.0192 |
| 20 | 0.5354 | 0.5317 | 0.0352 |
| 50 | 0.8241 | 0.7267 | 0.2800 |
| 75 | 0.9828 | 0.8550 | 0.5987 |
| 100 | 1.000 | 0.8950 | 0.7305 |

分析：

原本是希望training data的數量跟整個宇集的比例保持相同，然而在2 digit中，這樣的嘗試會讓data數過少，導致model根本學不到東西。

因此training data數就訂為原來3 digit的十分之一。

然而實際上這樣的data數量已經佔了超過整個宇集data數的一半以上，是個很不合理的比例，對此我猜測model想要歸納出規律，應該需要一定的data數量才可以。

4 bit digit

Add & Sub

Training set : 18000

Valid set : 2000

Testing set : 60000

|  |  |  |  |
| --- | --- | --- | --- |
| The number of training epochs | Train accuracy | Valid accuracy | Testing accuracy |
| 10 | 0.5592 | 0.5606 | 0.0390 |
| 20 | 0.6203 | 0.5967 | 0.0543 |
| 50 | 0.9595 | 0.6824 | 0.1361 |
| 75 | 1.0000 | 0.6846 | 0.1396 |
| 100 | 1.0000 | 0.6833 | 0.1385 |

4 bit digit

Add & Sub

Training set : 36000

Valid set : 4000

Testing set : 40000

|  |  |  |  |
| --- | --- | --- | --- |
| The number of training epochs | Train accuracy | Valid accuracy | Testing accuracy |
| 10 | 0.8861 | 0.8833 | 0.6098 |
| 20 | 0.9829 | 0.9595 | 0.8428 |
| 50 | 1.0000 | 0.9860 | 0.9499 |
| 75 | 1.0000 | 0.9885 | 0.9550 |
| 100 | 1.0000 | 0.9885 | 0.9570 |

分析：

對4 digit來說，18000個training data有些少了，model的結果出現了overfitting的現象，因此我把training data的數量增加為原來的2倍以後，準確率就明顯地往上升了。

同時，因為在和宇集的比例上，4 digit比3 digit少非常多，因此這樣的結果也可以用來當作證明「model想要歸納出規律，需要一定的data數量」的證據。

3 bit digit

Mul

Training set : 18000

Valid set : 2000

Testing set : 60000

|  |  |  |  |
| --- | --- | --- | --- |
| The number of training epochs | Train accuracy | Valid accuracy | Testing accuracy |
| 10 | 0.5047 | 0.5029 | 0.0254 |
| 20 | 0.9322 | 0.6163 | 0.0638 |
| 50 | 0.9997 | 0.6493 | 0.1018 |
| 75 | 1.0000 | 0.6468 | 0.0996 |
| 100 | 1.0000 | 0.6480 | 0.0980 |

3 bit digit

Mul

Training set : 54000

Valid set : 6000

Testing set : 20000

|  |  |  |  |
| --- | --- | --- | --- |
| The number of training epochs | Train accuracy | Valid accuracy | Testing accuracy |
| 10 | 0.6982 | 0.6904 | 0.1353 |
| 20 | 0.8056 | 0.7438 | 0.2261 |
| 50 | 0.9871 | 0.7516 | 0.2415 |
| 75 | 0.9995 | 0.7493 | 0.2469 |
| 100 | 1.0000 | 0.7556 | 0.2499 |

分析：

在multiply的情況比較奇妙，training 的時候出現明顯的overfitting現象，就算提高training data的數量也一樣，我猜測可能是因為這個model的架構並不符合乘法的規則，因為乘法是每個位數乘完之後，再全部加起來，這意味著低位跟高位會互相影響。