DSAI HW2 Adder & Subtractor

學號：P76071218 姓名：陳昱霖

Usage

```
$ python main.py [-t CAL_TYPE] [-d DIGITS] [-m MODEL_TYPE]
```

Options	Description
Calculation type	**Default: -t add**
-t add	addition
-t sub	subtraction
-t add_sub	addition & subtraction
-t mul	multiplication
Number of digits	**Default: -d 3**
-d 2	two digits
-d 3	three digits
-d 4	four digits
Model type	**Default: -t none**
-m add	addition model
-m sub	subtraction model
-m add_sub	addition & substraction model
-m mul	multiplication model

Model Structure

Layer (type)	Output Shape	Param #
lstm_1 (LSTM)	(None, 128)	73216
repeat_vector_1	(RepeatVecto (None, 4, 128)	0
lstm_2 (LSTM)	(None, 4, 128)	131584
time_distributed_1	(TimeDist (None, 4, 14)	1806

Total params: 206,606
Trainable params: 206,606
Non-trainable params: 0

Analysis

There are two part in this section.

- In the first part, comparing the performance of different type of calculation. Including addition, subtraction, mixed addition and subtraction, multiplition. In each model, training with two different size of dataset:

1. Larger Training set: Training data : 45000, Validation data: 5000, Testing data: 30000
2. Smallerer Training set: Training data : 18000, Validation data: 2000, Testing data: 10000

- In the second part, comparing the accuarcy of different size of digits (2,3 or 4) with the same type of calculation (mixed addition and subtraction)

Same digits (3 digits), different type of calculation

1. Adder

a. Larger Training set

- Data size:
  - Training data : 45000
  - Validation data : 5000
  - Testing data : 30000
  - Iteration: 50

```
$ python main.py -t add -d 3
```

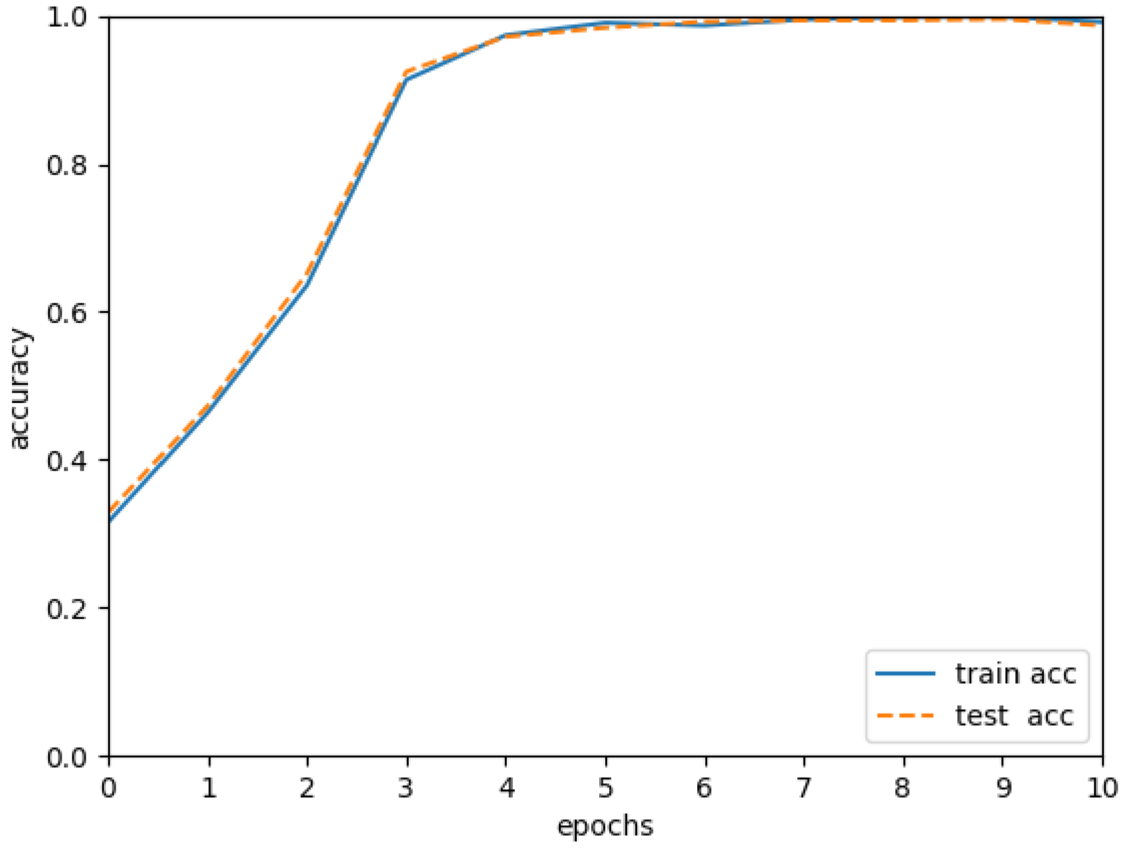
- Result

```
Iteration 1
Q 434+5   T 439  ☒ 26
Q 183+712 T 895  ☒ 106
Q 74+193  T 267  ☒ 106
Q 690+51  T 741  ☒ 106
Q 778+874 T 1652 ☒ 1100
Q 65+32   T 97   ☒ 66
Q 918+346 T 1264 ☒ 1140
Q 60+113  T 173  ☒ 106
Q 22+750  T 772  ☒ 666
Q 525+2   T 527  ☒ 26
...
Iteration 50
Q 6+939   T 945  ☑ 945
Q 190+425 T 615  ☑ 615
Q 87+142  T 229  ☑ 229
Q 705+62  T 767  ☑ 767
Q 246+628 T 874  ☑ 874
Q 93+537  T 630  ☑ 630
Q 968+51  T 1019 ☒ 1029
Q 940+63  T 1003 ☑ 1003
Q 275+403 T 678  ☑ 678
Q 212+915 T 1127 ☑ 1127
```

Iteration	**Training Loss**	**Training Accuracy**	**Validation Loss**	**Validation Accuarcy**
1	1.9217	0.3153	1.8286	0.3286
10	0.9795	0.636	0.9396	0.649
20	0.1262	0.9736	0.1241	0.9699
30	0.0509	0.9868	0.0368	0.9913
40	0.0117	0.9983	0.0214	0.9937
50	0.0282	**0.9916**	0.0408	0.9863

```
Testing loss: 0.0418
Testing acc: 0.9867
```

- Accuracy of Training and Testing data for each epoch



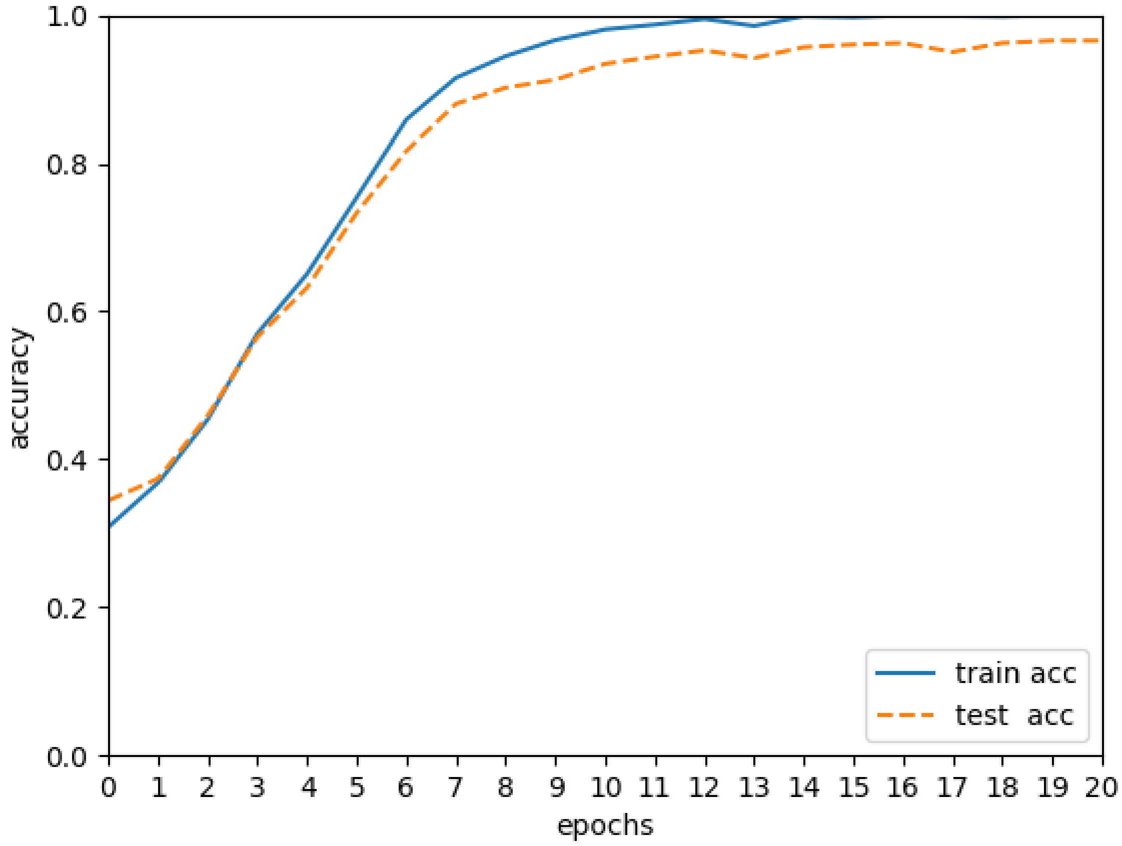
b. Smaller Training set

- Data size:
  - Training data : 18000
  - Validation data : 2000
  - Testing data : 10000
  - Iteration: 100
- Result

Iteration	Training Loss	Training Accuracy	Validation Loss	Validation Accuarcy
1	1.9991	0.3078	1.8347	0.3356
10	1.4703	0.4539	1.4519	0.4571
20	0.9413	0.6507	0.98	0.6272
30	0.443	0.8589	0.5194	0.8076
40	0.2088	0.9446	0.2934	0.9001
50	0.0989	0.9807	0.1932	0.935
60	0.0468	0.995	0.1333	0.9539
70	0.024	0.9988	0.12	0.9569
80	0.0124	0.9999	0.1057	0.9643
90	0.0148	0.9984	0.1035	0.9664
100	0.0044	1	0.0942	0.9682

```
Testing loss: 0.0988
Testing acc: 0.9656
```

- Accuracy of Training and Testing data for each epoch



2. Subtractor

a. Larger Training set

- Data size:
  - Training data : 45000
  - Validation data : 5000
  - Testing data : 30000
  - Iteration: 50

```
$ python main.py -t sub -d 3
```

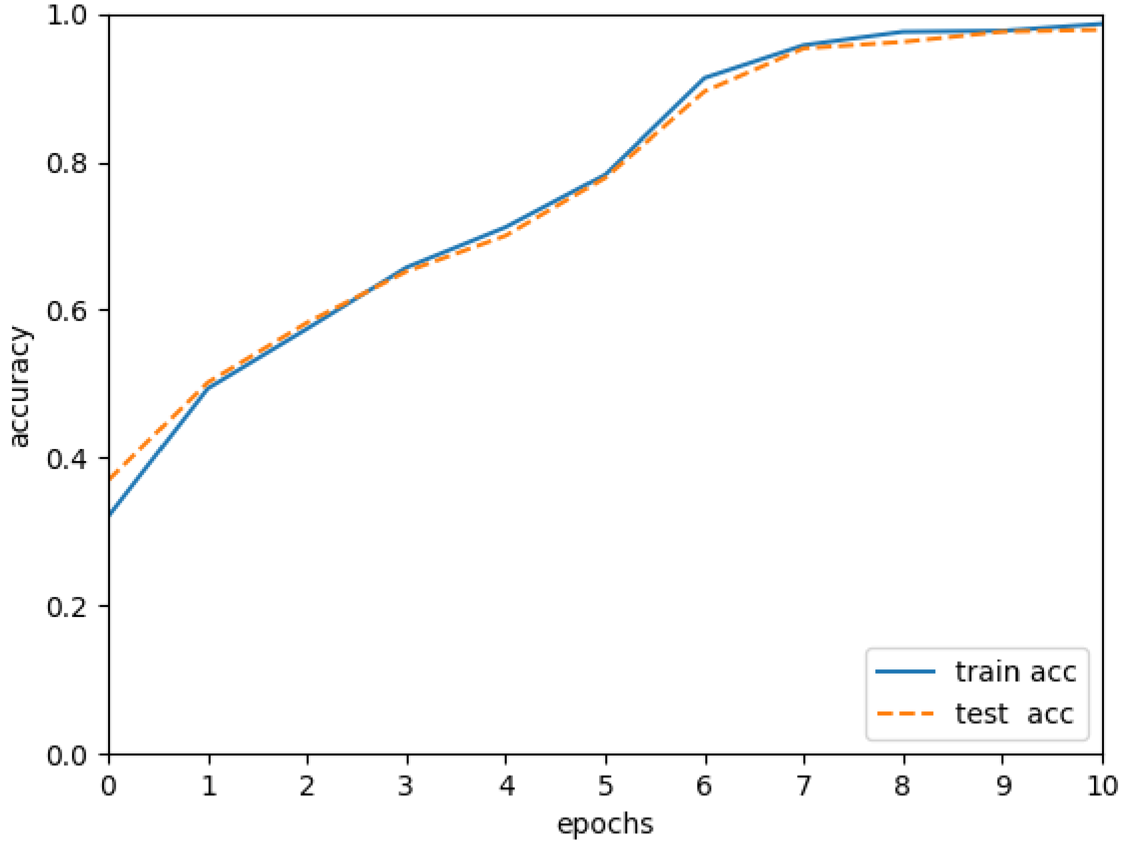
- Result

```
Iteration 1
Q 36-166  T -130 ☒ -112
Q 363-907 T -544 ☒ -10
Q 536-198 T 338  ☒ -10
Q 852-124 T 728  ☒ 110
Q 841-99  T 742  ☒ 14
Q 74-53   T 21   ☒ 11
Q 260-63  T 197  ☒ 11
Q 647-872 T -225 ☒ -10
Q 712-129 T 583  ☒ 11
Q 427-918 T -491 ☒ -103
...
Iteration 50
Q 87-541  T -454 ☑ -454
Q 596-73  T 523  ☑ 523
Q 63-169  T -106 ☑ -106
Q 371-435 T -64  ☑ -64
Q 83-846  T -763 ☑ -763
Q 900-210 T 690  ☑ 690
Q 72-931  T -859 ☑ -859
Q 628-581 T 47   ☑ 47
Q 7-129   T -122 ☑ -122
Q 292-391 T -99  ☒ -19
```

Iteration	Training Loss	Training Accuracy	Validation Loss	Validation Accuarcy
1	1.9741	0.3144	1.7215	0.3648
10	1.1594	0.5714	1.1365	0.5766
20	0.8028	0.7069	0.8057	0.7064
30	0.3334	0.8937	0.3394	0.8882
40	0.1027	0.9734	0.1281	0.9581
50	0.0502	**0.989**	0.0686	0.9805

```
Testing loss: 0.0676
Testing acc: 0.9804
```

- Accuracy of Training and Testing data for each epoch



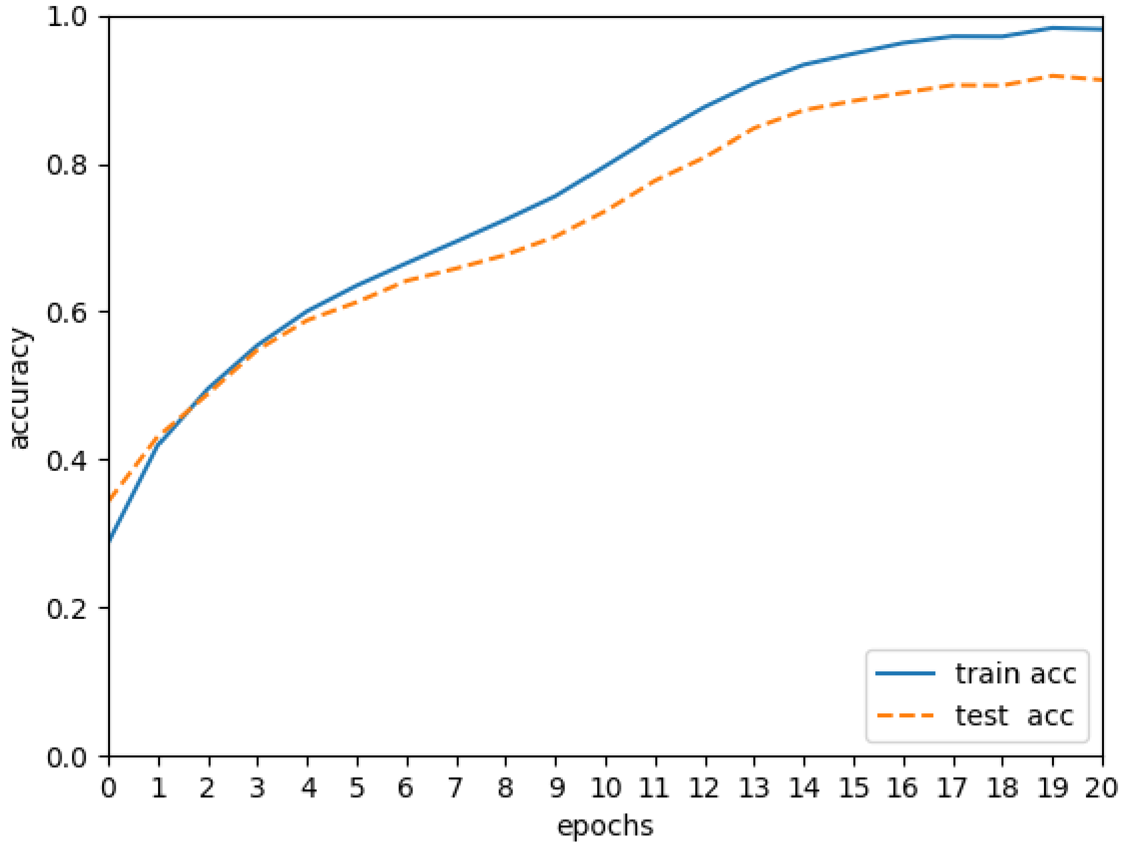
b. Smaller Training set

- Data size:
  - Training data : 18000
  - Validation data : 2000
  - Testing data : 10000
  - Iteration: 100
- Result

Iteration	Training Loss	Training Accuracy	Validation Loss	Validation Accuarcy
1	2.1543	0.287	1.8822	0.349
10	1.3706	0.4955	1.3782	0.4844
20	1.0971	0.6005	1.1144	0.5924
30	0.9169	0.6653	0.9571	0.6426
40	0.7589	0.7242	0.8505	0.6736
50	0.5657	0.7966	0.6593	0.7396
60	0.3755	0.8759	0.4975	0.8065
70	0.2398	0.9331	0.3666	0.8686
80	0.1557	0.9625	0.3017	0.892
90	0.1161	0.9711	0.2677	0.9044
100	0.0846	0.981	0.254	0.9119

```
Testing loss: 0.2537
Testing acc: 0.9126
```

- Accuracy of Training and Testing data for each epoch



3. Mixed Adder and Subtractor

a. Larger Training set

- Data size:
    - Training data : 45000
    - Validation data : 5000
    - Testing data : 30000
    - Iteration: 50

```
$ python main.py -t add_sub -d 3
```

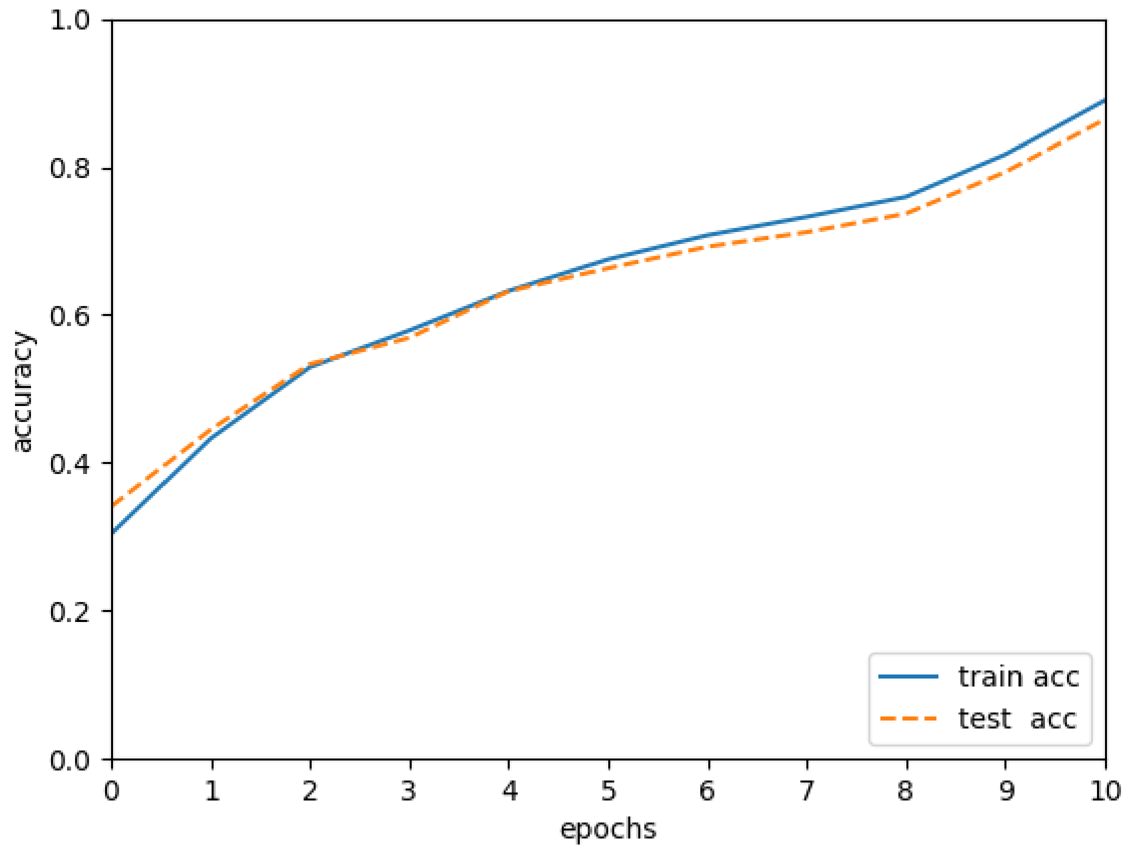
- Result

```
Iteration 1
Q 242-695 T -453 ☒ -11
Q 415+345 T 760  ☒ 105
Q 440+430 T 870  ☒ 101
Q 62+616  T 678  ☒ 105
Q 57-556  T -499 ☒ -115
Q 937-53  T 884  ☒ 13
Q 75+58   T 133  ☒ 158
Q 161+34  T 195  ☒ 158
Q 734-332 T 402  ☒ 111
Q 523+87  T 610  ☒ 105
...
Iteration 50
Q 147-5   T 142  ☒ 132
Q 82-894  T -812 ☑ -812
Q 71+320  T 391  ☑ 391
Q 97+334  T 431  ☑ 431
Q 296+782 T 1078 ☒ 1089
Q 949-37  T 912  ☒ 913
Q 697-803 T -106 ☒ -117
Q 3-815   T -812 ☑ -812
Q 401+775 T 1176 ☑ 1176
Q 601-49  T 552  ☑ 552
```

Iteration	**Training Loss**	**Training Accuracy**	**Validation Loss**	**Validation Accuarcy**
1	1.9715	0.3037	1.7849	0.3439
10	1.2731	0.5294	1.2576	0.5318
20	0.9835	0.6324	0.9835	0.631
30	0.7903	0.7077	0.8207	0.6894
40	0.6514	0.7597	0.7056	0.7309
50	0.3221	**0.8896**	0.3768	0.8611

```
Testing loss: 0.3694
Testing acc: 0.8641
```

- Accuracy of Training and Testing data for each epoch



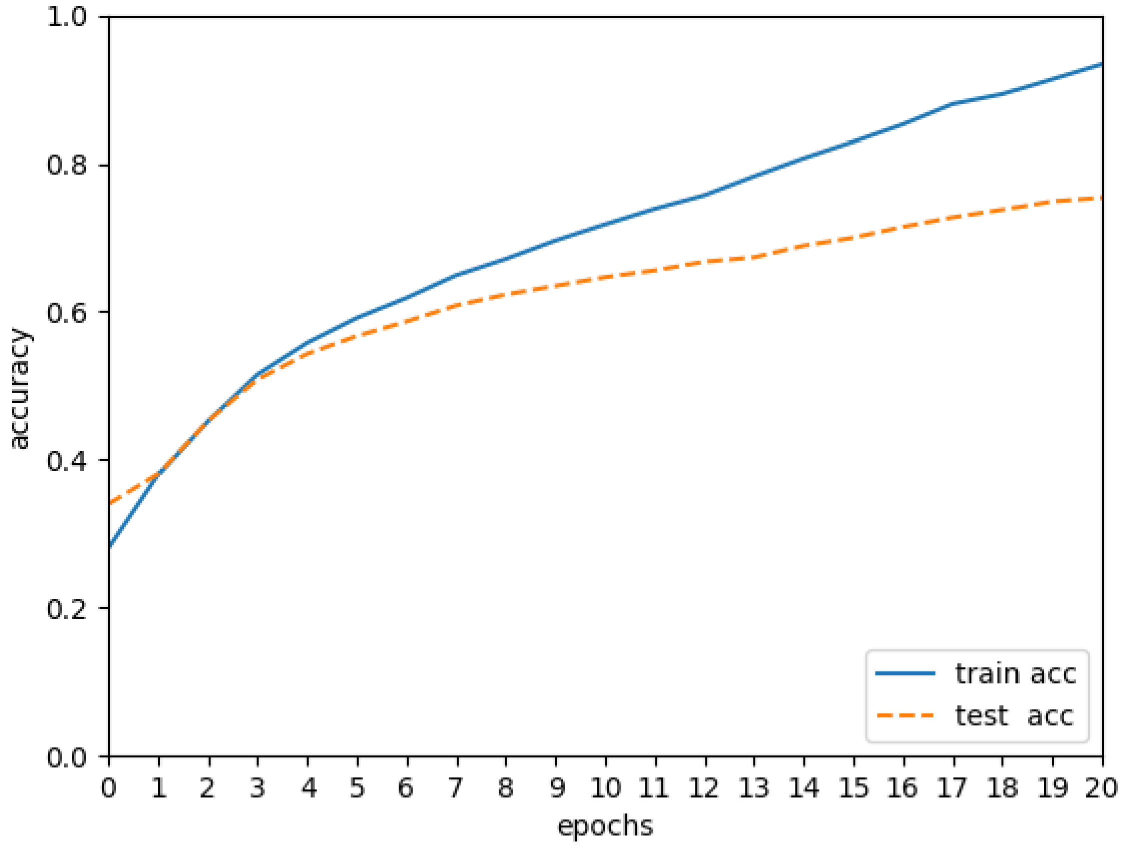
b. Smaller Training set

- Data size:
  - Training data : 18000
  - Validation data : 2000
  - Testing data : 10000
  - Iteration: 100
- Result

Iteration	Training Loss	Training Accuracy	Validation Loss	Validation Accuarcy
1	2.12	0.2798	1.8705	0.3374
10	1.4887	0.4519	1.4843	0.4485
20	1.2018	0.558	1.2303	0.5433
30	1.0386	0.6186	1.0907	0.5915
40	0.9053	0.6712	0.9959	0.6237
50	0.7879	0.7178	0.9417	0.6474
60	0.6768	0.7571	0.8809	0.6704
70	0.5497	0.807	0.8065	0.6879
80	0.4302	0.8528	0.7492	0.7115
90	0.3313	0.8934	0.6916	0.7374
100	0.2409	0.9338	0.6805	0.7526

```
Testing loss: 0.6715
Testing acc: 0.7539
```

- Accuracy of Training and Testing data for each epoch



4. Multiplier

a. Larger Training set

- Data size:
    - Training data : 45000
    - Validation data : 5000
    - Testing data : 30000
    - Iteration: 50

```
$ python main.py -t mul -d 3
```

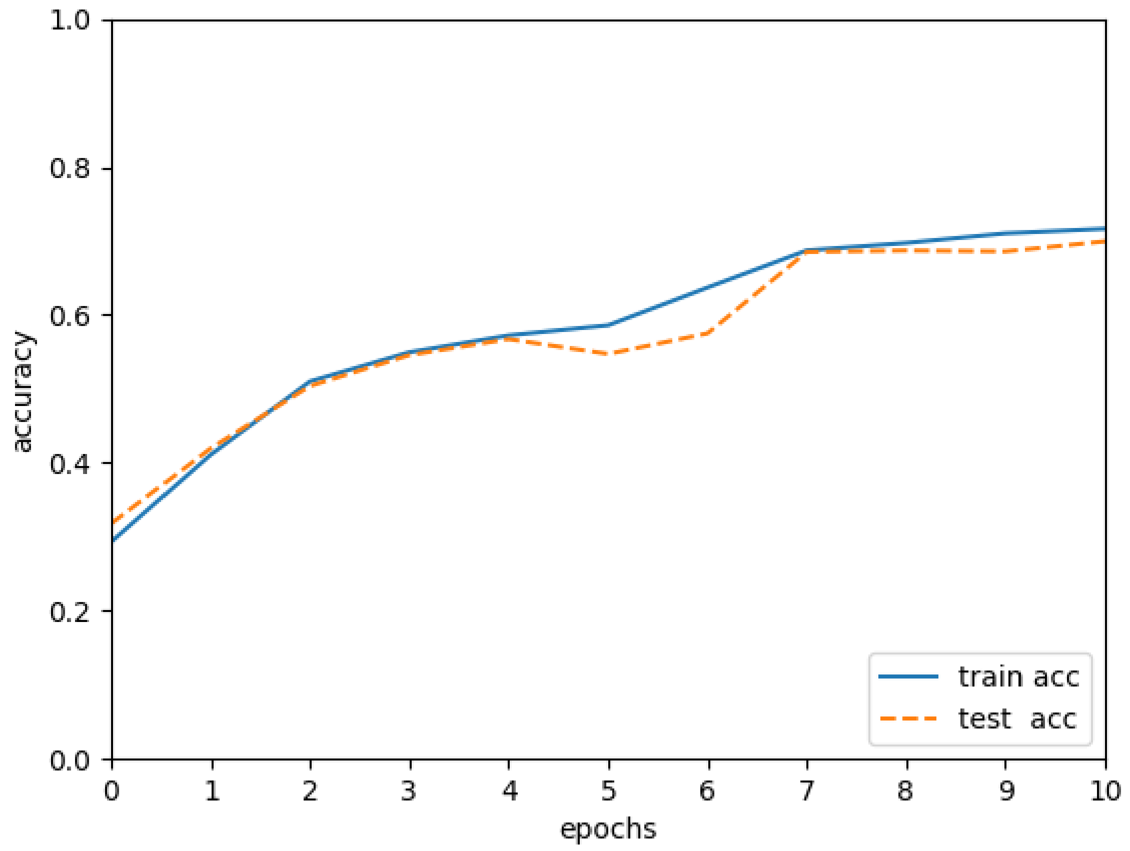
- Result

```
Iteration 1
Q 313*23  T 7199   ☒ 1199
Q 454*78  T 35412  ☒ 13666
Q 41*815  T 33415  ☒ 1155
Q 587*388 T 227756 ☒ 139662
Q 4*534   T 2136   ☒ 1122
Q 166*595 T 98770  ☒ 155555
Q 227*3   T 681    ☒ 116
Q 201*19  T 3819   ☒ 1190
Q 740*772 T 571280 ☒ 119660
Q 91*24   T 2184   ☒ 116
...
Iteration 50
Q 166*43  T 7138   ☒ 7518
Q 347*284 T 98548  ☒ 98004
Q 647*89  T 57583  ☒ 57613
Q 90*401  T 36090  ☑ 36090
Q 497*88  T 43736  ☒ 43116
Q 72*10   T 720    ☑ 720
Q 71*452  T 32092  ☒ 31852
Q 889*545 T 484505 ☒ 482245
Q 113*2   T 226    ☑ 226
Q 6*883   T 5298   ☒ 5258
```

Iteration	Training Loss	Training Accuracy	Validation Loss	Validation Accuarcy
1	1.9295	0.2928	1.8293	0.3164
10	1.2217	0.51	1.2141	0.5075
20	1.0571	0.5722	1.0586	0.5694
30	0.9443	0.637	1.0645	0.5738
40	0.7755	0.6973	0.7901	0.6894
50	0.7193	**0.7165**	0.7458	0.6988

```
Testing loss: 0.7455
Testing acc: 0.6995
```

- Accuracy of Training and Testing data for each epoch



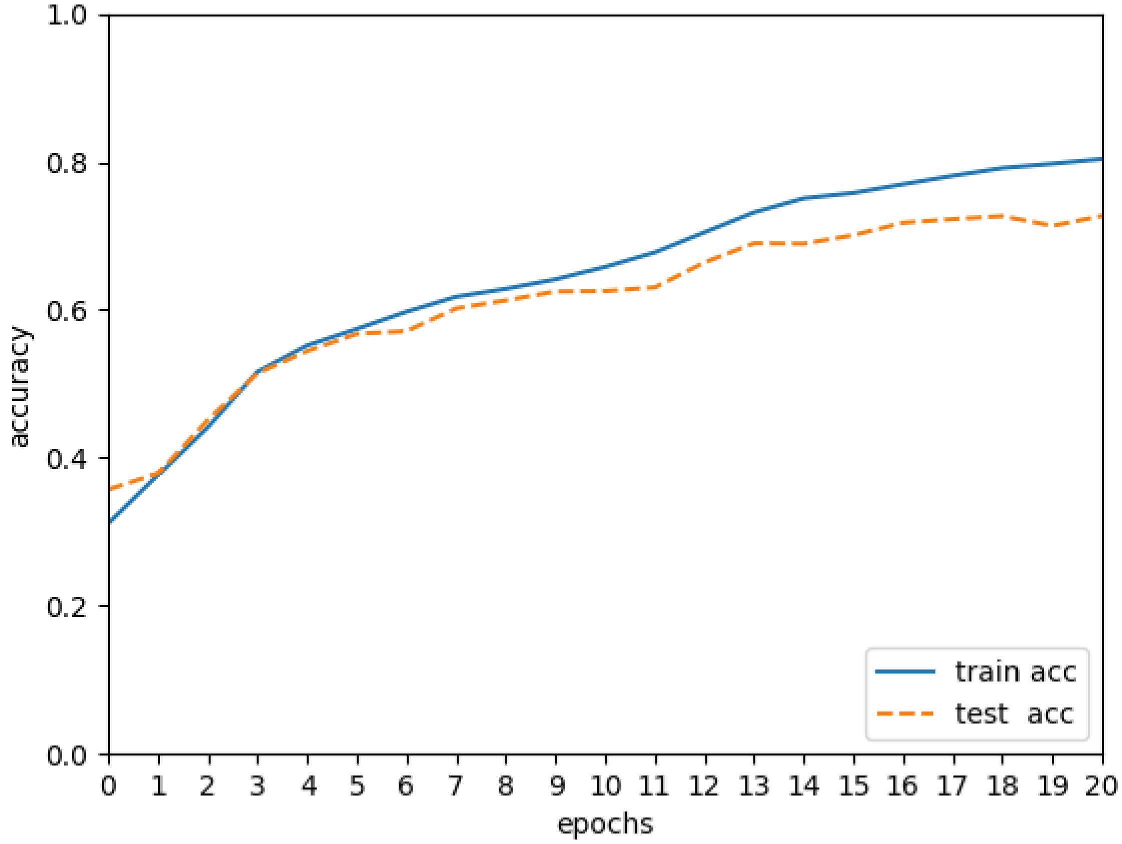
b. Smaller Training set

- Data size:
  - Training data : 18000
  - Validation data : 2000
  - Testing data : 10000
  - Iteration: 100
- Result

Iteration	Training Loss	Training Accuracy	Validation Loss	Validation Accuarcy
1	1.9544	0.3108	1.7724	0.354
10	1.4444	0.4415	1.4279	0.4476
20	1.1297	0.552	1.1353	0.5447
30	1.0204	0.5974	1.0536	0.5726
40	0.9435	0.6285	0.9688	0.6131
50	0.8823	0.6582	0.9397	0.6278
60	0.7963	0.705	0.8692	0.6618
70	0.6893	0.751	0.8146	0.6913
80	0.6305	0.77	0.7489	0.7185
90	0.5762	0.7919	0.7312	0.7238
100	0.5424	0.8041	0.7397	0.7277

```
Testing loss: 0.7384
Testing acc: 0.7272
```

- Accuracy of Training and Testing data for each epoch



Same type of calculation (Add_Sub), different number of digits (2,3,4 digits)

1. 2 digits

```
$ python main.py -t add_sub -d 2
```

Iteration	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
1	1.6718	0.4373	1.4028	0.4889
10	0.3969	0.8852	0.3763	0.8915
20	0.092	0.9805	0.0916	0.9803
30	0.0293	0.9954	0.0373	0.9911
40	0.022	0.9955	0.0188	0.9959
50	0.0057	**0.9996**	0.0122	0.9972

Testing loss: 0.0133
Testing acc: 0.9964

2. 3 digits

```
$ python main.py -t add_sub -d 3
```

Iteration	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
1	1.9715	0.3037	1.7849	0.3439
10	1.2731	0.5294	1.2576	0.5318
20	0.9835	0.6324	0.9835	0.631
30	0.7903	0.7077	0.8207	0.6894
40	0.6514	0.7597	0.7056	0.7309
50	0.3221	**0.8896**	0.3768	0.8611

```
Testing loss: 0.3694
Testing acc: 0.8641
```

3. 4 digits

```
$ python main.py -t add_sub -d 4
```

Iteration	Training Loss	Training Accuracy	Validation Loss	Validation Accuarcy
1	1.9156	0.3233	1.7493	0.3579
10	1.1699	0.5604	1.164	0.5609
20	0.9303	0.6505	0.9685	0.6341
30	0.7762	0.7079	0.8447	0.675
40	0.62	0.7658	0.711	0.7196
50	0.4641	**0.8297**	0.5922	0.7713

```
Testing loss: 0.6005
Testing acc: 0.7699
```

Conclusion

Different type of calculation:

When training by a smaller dataset, most of the model got a lower accuarcy, except multiplication.

(Training: 45000, validation: 5000, testing: 30000, Iteration: 50)

1. Addition (Acc: 0.9867)
2. Subtraction (Acc: 0.9804)
3. Addition & Subtraction (Acc: 0.8641)
4. Multiplication (Acc: 0.6995)

(Training: 18000, validation: 2000, testing: 10000, Iteration: 100)

1. Addition (Acc: 0.9656)
2. Subtraction (Acc: 0.9126)
3. Addition & Subtraction (Acc: 0.7539)
4. Multiplication (Acc: 0.7272)

Different number of digit:

When applying fewer number of digit, the accuarcy increased.

(Training: 45000, validation: 5000, testing: 30000, Iteration: 50)

1. 2 digits (Acc: 0.9964)
2. 3 digits (Acc: 0.8641)
3. 4 digits (Acc: 0.7699)