

Author: Xing Qian

UO ID:951649469

link to your Google Drive Folder:

<https://drive.google.com/drive/folders/1rK0iLQdUTJVolqKyVFNZC0CJt0CMjOFM?usp=sharing>

link to your Google Colab File:

[https://drive.google.com/file/d/1q0jeDDe6WaX2Wh30tiF-UUAR95aIJ\\_jK/view?usp=sharing](https://drive.google.com/file/d/1q0jeDDe6WaX2Wh30tiF-UUAR95aIJ_jK/view?usp=sharing)

Credits: None

## Report

Data is the most important factor affecting the results of the model. Therefore, I decided to use the data set glove.6b.300d.txt, which contains a large embedding dimension and can represent different words more discretely. This can provide the quality of data and improve the accuracy of the model. Also set const\_ MAX\_ Length = 300, because the maximum text length in the dataset is 300

Here we use the simplest method, that is, adjust the parameter value of one parameter at a time, and so on, until all the optimal parameters are found. Although this operation process may not necessarily find the optimal parameters, to a certain extent, it allows us to find relatively acceptable parameters. The parameter selection process is as follows:

(1) Change the dropout value when hidden size = 64, kernel size=3, epoch=20, lr=0.01, batch size=32 and optimizer=SGD

Table 1 different dropout value VS train and dev accuracy

value	Train Accuracy	Dev accuracy
0.2	0.9448	0.8443
0.3	0.9242	0.8428
0.4	0.9048	0.8572
0.5	0.8819	0.8498

It can be seen that when dropout is equal to 0.4, the result of dev accuracy is the best, so the optimal index of dropout is 0.4.

(2) Change hidden size value when dropout=0.4, kernel size=3, epoch=20, lr=0.01, batch size=32 and optimizer=SGD

Table 2 different hidden size VS train and dev accuracy

value	Train Accuracy	Dev accuracy
64	0.9013	0.8541
128	0.9237	0.8561
256	0.9427	0.8608
512	0.9534	0.8611

As you can see, with the increase of the size of hidden size, the value of dev occurrence increases gradually. Finally, the optimal hidden size = 512 can be determined.

(3) Change kernel size value when dropout=0.4, hidden size=512, epoch=20, lr=0.01, batch size=32 and optimizer=SGD

Table 3 different kernel size VS train and dev accuracy

value	Train Accuracy	Dev accuracy
3	0.9524	0.8220
5	0.9798	0.8596
7	0.9881	0.8466
9	0.9932	0.8600
11	0.9904	0.8544

It can be found from the above that when the kernel size is equal to 9, the result is better. Therefore, the value of this super parameter can be determined as 9

(4) Change epoch value when dropout=0.4, hidden size=512, kernel size=9, lr=0.01, batch size=32 and optimizer=SGD

Table 4 different epoch VS train and dev accuracy

value	Train Accuracy	Dev accuracy
20	0.9934	0.8552
40	0.9969	0.8594
60	0.9990	0.8609
80	0.9983	0.8591
100	0.9993	0.8630
120	0.9993	0.8343
140	0.9996	0.8608
160	0.9997	0.8611
180	0.9994	0.8594

Through the comparison in the above table, 100 is the best choice for epoch. However, by observing the output of each epoch, we can find that the accuracy has been constantly fluctuating, and there may be over fitting problems. At the same time, considering the time cost and the influence of random fluctuation in the process of model training, we can think that epoch is in these value ranges, except 120, the result can be considered as approximately equivalent, so here I set epoch to 40 to weigh time and accuracy.

(5) Change lr value when dropout=0.4, hidden size=512, kernel size=9, epoch=100, batch size=32 and optimizer=SGD

Table 4 different lr VS train and dev accuracy

value	Train Accuracy	Dev accuracy
0.01	0.9988	0.8596
0.001	0.9900	0.8588
0.0001	0.5596	0.6202

According to the above table, I choose to set LR to 0.01.

(6) Change batch size when dropout=0.4, hidden size=512, kernel size=9, epoch=100, lr=0.01 and optimizer=SGD

Table 5 different batch size VS train and dev accuracy

value	Train Accuracy	Dev accuracy
32	0.9988	0.7768

<b>64</b>	<b>0.9995</b>	<b>0.8592</b>
<b>128</b>	<b>0.9993</b>	<b>0.8613</b>

According to table 5, select batch size 128.

(7) Change optimizer value when dropout=0.4, hidden size=512,kernel size=9,epoch =100,lr=0.01 and batch size=128

Table 6 different optimizer VS train and dev accuracy

<b>value</b>	<b>Train Accuracy</b>	<b>Dev accuracy</b>
<b>SGD</b>	<b>0.9993</b>	<b>0.8470</b>
<b>Adam</b>	<b>0.9278</b>	<b>0.8280</b>
<b>Adadelata</b>	<b>0.9990</b>	<b>0.8617</b>

Therefore, adadelata optimizer is selected to optimize the model parameters.

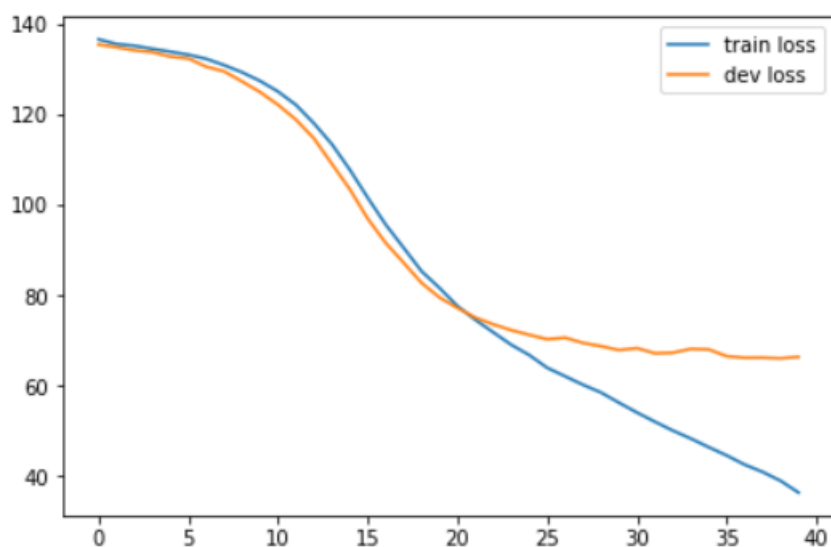
(8) According to all the above testing process, the optimal parameter combination is finally selected, as shown in Table 7

Table 7 Final parameters select

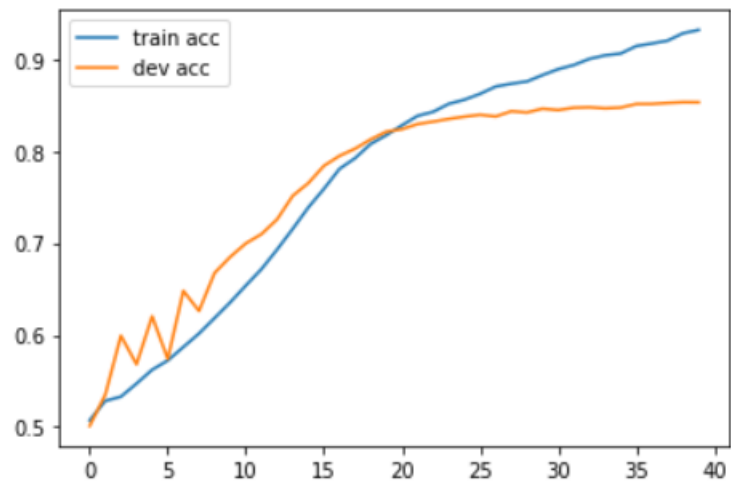
<b>name</b>	<b>value</b>	<b>Name2</b>	<b>value</b>
<b>dropout</b>	<b>0.4</b>	<b>epoch</b>	<b>40</b>
<b>hidden_size</b>	<b>512</b>	<b>lr</b>	<b>0.01</b>
<b>kernel_size</b>	<b>9</b>	<b>batch_size</b>	<b>128</b>
<b>optimizer</b>	<b>Adadelata</b>		

Training progress visualization

When using the best parameters for training, the value of dev ACC is 0.8532.



Picture 1 loss VS epoch



Picture 2 accuracy VS epoch