Hyperparameter Optimazation of Convolutional Neural Networks on Digit Recognition Task

Yang Li[1]

I. Strategy

There are five hyperparameters I need for this experiment in Convolutionalnetwork under NeuralNetworks directory. To achieve the best possible testing accuracy, I want to optimize the five hyperparameters, and the key is to find a optimal value for each parameter by changing the value of a parameter while keeping the values of other four unchanged. By doing so I obtain optimal values for the five parameters and I can test the combinations of the optimal values of the parameters expecting to achieve the best possible testing accuracy.

II. Experiment

A. Learning Rate

From Network hyperparamters table (Tobias), learning rate is set to 0.001 by default and tested from 0.0000001 to 0.5. For the sake of time and effort, I cannot run one million times of this experiment, so I select learning rates, by default 0.001, from 0.00001 to 0.5. Here is the way I select them: the second is one fifth as big as the first one, and the third is a half of the second, and the fourth is one fifth as big as the third, and so. (0.5, 0.1, 0.05, 0.01 .... 0.00001)

From 0.5 to 0.05, testing accuracy increases a little as learning rate decreases. The reason may be our learner has left out too much important informaiton as it learn through our dataset. From 0.05 to 0.01, testing accuracy dramatically increases as learning rate decreases. The reason may be our learner has learnt something critical from the training set which helps on identifying digits in the testing set. Testing accuracy reaches the peak, 0.960938, when learning rate decreases to 0.05 and then dramatically drops as learning decreases. The reason comes from overfitting. Our learner has found out a lot of characters critical to identifying digits but pays extra attention to details which do not help at all.

B. Training Iterations

Training iterations is set to 10000 by default. I start testing it from 500 and double it in the second time and quadruple it in the third time. (500, 1000, 2000 ... 128000) I achieve pretty decent testing accuracy by default so I wonder how setting training iterations would affect testing accuracy. After I figure out what number of training iterations would yield optimal testing accuracy, I will also set the number of training iterations to that number and also set learning rate to 0.05 and see if two optimal hyperparameters will output a testing accuracy greater than 0.960938.

*This work was not supported by any organization

As training iteration increases, testing accuracy increases as well. From 500 to 8000, testing accuracy dramatically increases as training iteration increases. The reason may be the more our learner trains the more important information it extracts while traing with the traing set and better it becomes on identifying digits on the testing set. From 8000 to 128000, testing accuracy slowly increases as traing iteration increases. The reason may be our learner has already gain important information from training a certain number of times, 30000 perhaps, and the more it trains and more difficult it finds out important information from the training set. I set learning rate to be 0.05 and training iterations to be 128000 and achieve the best ever testing accuracy 0.988281 which tells us optimizing learning rate does not conflicts with optimizing training iterations.

C. Batch Size

Batch size is set to 128 by default. Because the default batch size is power of two, I want to test batch sizes in power of two as well. Starting the sizes smaller than the default size, I have 2, 4, 8, 16, 32, 64. Depending on what I find out the correlation between batch size fewer than 128 and testing accuracy, I may test on batch sizes greater than 128. As always, if I happen to find out a optimal batch size, I want to combine it with the optimal training iterations and learning rate, and test the combined hyperparameters to see if I can obtain a better testing accuracy.

As batch size increases from 2 to 64, testing accuracy does not have obvious increase or decrease: with batch size of 2 testing accuracy is lower than it is with batch size of 4 as if testing accuracy increased as batch size increased, but testing accuracy with batch size of 16 is lower than it is with batch size of 8 as testing accuracy increase as batch size dropped where we would have a contradition. I test on batch size of 256 and achieve a worse. Given no more data, Whether increasing the number of batch size greater than 128 lower testing accuracy has remained unknown and need to be explored in the future.

D. Display Step

The number of display steps is set to 10 by default. At first, I want to test how testing accuracy changes as the number of display steps increases every 10 steps. I have found out no obvious change on testing accuracy (0.85, 0.88, 0.83, 0.84). I guess it may be because the number of steps increased is not large enough. I set the number of display steps to be 100 and achieve still no obvious change on testing accuracy (0.88). Increasing the number of display steps has become pointless,

but I still want to know how testing accuracy changes when the number of display steps decreases from 10, the default value.

There are only 9 (1 - 9) possible positive values of display step fewer than default to test on, but I test on four consecutive values 5 to 8. I observe no obvious change to testing accuracy as the number of steps increase one by one from 5 to 8.

E. Dropout

The number of Dropout is set to 0.75 by default. We have observed how testing accuracy changes as dropout increases from the homework. From 0 to 1, testing accuracy increases as dropout increases and it drops as dropout increases beyond 1. It turns out our obeservation from the homework totally applies to my testing, dispite some random error I get, testing accuracy with dropout 0.95 is greater than testing accuracy with dropout 0.97.

After finding out a optimal value of dropout 0.95, I test it with values of two other hyperparameters, learning rate 0.05 and training iteration 64000, and I achieve the same best testing accuracy 0.988281. If I had tested it with the optimal value of training iterations 128000, I would have achieved the best ever testing accuracy, greater than 0.988281 for sure.

III. Conclusions

By optimizing each of the five parameter, I have found out testing accuracy has a turning point as each of the values for learning rate and dropout increases. Testing accuracy has positive corelation with the number of training iterations and no obvious corelation with display step. The corelation between testing accuracy and batch size is not very clear and need to be explored in the future particularly the batch sizes greater than the default 128.

The combination of optimized hyperparameters, which are learn rate 0.05, training iterations 64000, batch size 128, display step 10, and dropout 0.95, I have tested gives me testing accuracy 0.988281. As I mentioned before, I am pretty confident the testing accuracy can be improved if the number of training iterations increases.

References

[1] G. O. Young, Synthetic structure of industrial plastics (Book style with paper title and editor), in Plastics, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 1564.

[2] W.-K. Chen, Linear Networks and Systems (Book style). Belmont, CA: Wadsworth, 1993, pp. 123135.

[3] H. Poor, An Introduction to Signal Detection and Estimation. New York: Springer-Verlag, 1985, ch. 4.

[4] B. Smith, An approach to graphs of linear forms (Unpublished work style), unpublished.

[5] E. H. Miller, A note on reflector arrays (Periodical styleAccepted for publication), IEEE Trans. Antennas Propagat., to be publised.

[6] J. Wang, Fundamentals of erbium-doped fiber amplifiers arrays (Periodical styleSubmitted for publication), IEEE J. Quantum Electron., submitted for publication.

[7] C. J. Kaufman, Rocky Mountain Research Lab., Boulder, CO, private communication, May 1995.