DD 2424 - Assignment 2 Bonus

Jia Fu (jiafu@kth.se)

April 28, 2021

Hidden Nodes

In this section, 45000 data points are used for training and 5000 data points for validation. `n_s = 2 * floor(n / n_batch)` and 2 cycles' training are implemented. Other hyper-parameters settings are: `n_batch = 100`, `eta_min = 1e-5`, and `eta_max = 1e-1`. Different numbers of hidden nodes are tested and coarse-to-fine random searches are done to set `lambda` for these networks. Table 1 shows the results:

Number of hidden nodes	25	50	100
Best Searched λ	0.000029	0.003143	0.004592
Validation Accuracy	49.50%	53.08%	54.28%
Test Accuracy	48.99%	52.18%	53.70%

Table 1: The impact of different numbers of hidden nodes on the network performance

It can be seen that with more hidden nodes the amount of regularization has to increase for the network to obtain its optimal performance. Within a certain range, the more hidden nodes, the higher the accuracy if we keep all other hyper-parameters of the network the same except for λ.

Dropout

If the network has a high number of hidden nodes which means more regularization, then the dropout strategy can be employed. Each hidden node has the probability p to be dropped out randomly in every update step. The network with 1000 hidden nodes is tested here. Retain other hyper-parameters used in the last section, p and λ settings and the corresponding test accuracy are shown in table 2.

	$p = 0, \lambda = 0$	$p = 10\%, \lambda = 0$	$p = 0, \lambda = 0.002$
Test Accuracy	55.58%	55.82%	55.80%

Table 2: The comparison of applying only regularization, only dropout, and neither

Obviously, dropout somehow can substitute regularization to compensate the overfitting when optimizing the performance of the network. In my experiment, either adding the regularization term where $\lambda = 0.002$ or dropping out around 10% hidden nodes randomly in each update step can improve the test accuracy by around 0.2% based on the performance of the network which applys neither of these two methods. **55.82%is also the highest test accuracy I found in different optimization approaches.**

More Search

More exhaustive random searches are done in this section to find optimal hyper-parameters. I use the network with 100 hidden nodes where the coarse-to-fine search gives us `lambda = 0.004592` in the first section. Firstly, the length of the cycles is searched in the range of 500 – 1500. Retain other hyper-parameters used in the first section, three best results are:

	n_s	Validation Accuracy	Test Accuracy
1	1470	55.00%	54.27%
2	1479	55.44%	54.12%
3	1485	55.34%	53.48%

Table 3: `n_s` settings for the 3 best performing networks from the random search

Then the number of the cycles is tested in the range of 1 – 10 orderly, this can be realized by changing `n_epoch`. Here I set `n_s = 1470` obtained in the last step, three best results are:

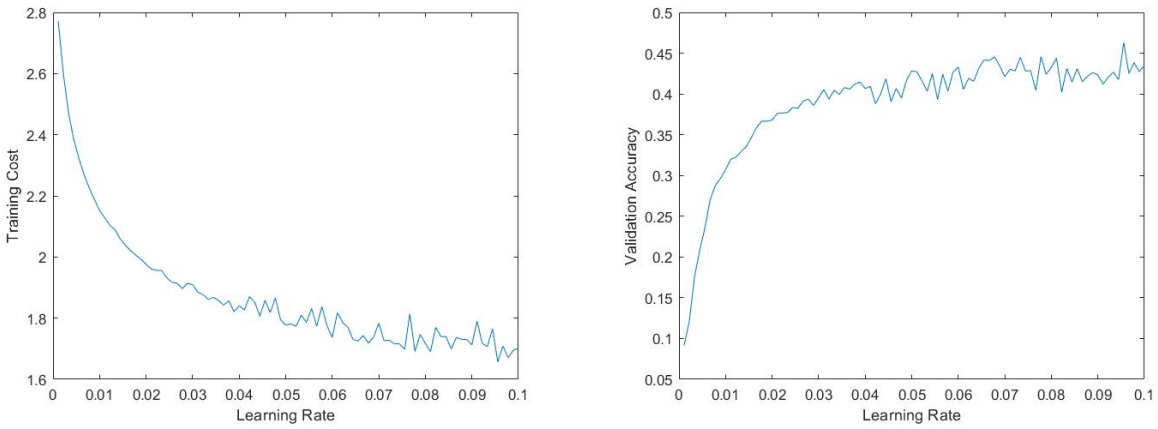
	Number of cycles	Validation Accuracy	Test Accuracy
1	6	56.42%	55.48%
2	9	55.42%	55.15%
3	10	55.42%	55.03%

Table 4: Number of cycles settings for the 3 best performing networks from the search

If the best searched `lambda`, `n_s`, and number of cycles are applied to the network with 100 hidden nodes, the test accuracy can exceed 55% which is comparable to the performance of the network with 1000 hidden nodes. **It can be seen from the above optimization strategies that the method of increasing hidden nodes has the most obvious effect on accuracy improvement.**

Learning Rate (LR) Range Test

In this section, "LR range test" [1] is implemented on the network with 50 hidden nodes. Still, 45000 data points are used for training and 5000 data points for validation. `n_batch = 100` so one epoch contains 450 update steps. Then `n_s = 900` and `n_epoch = 2` are set to make the training go through half a cyclical learning rates' cycle. This is to ensure that the learning rate increases linearly between low and high boundary values which are set by `eta_min = 0` and `eta_max = 1e-1` here respectively. $\lambda = 0.003143$ which was obtained by previous random search is applied.



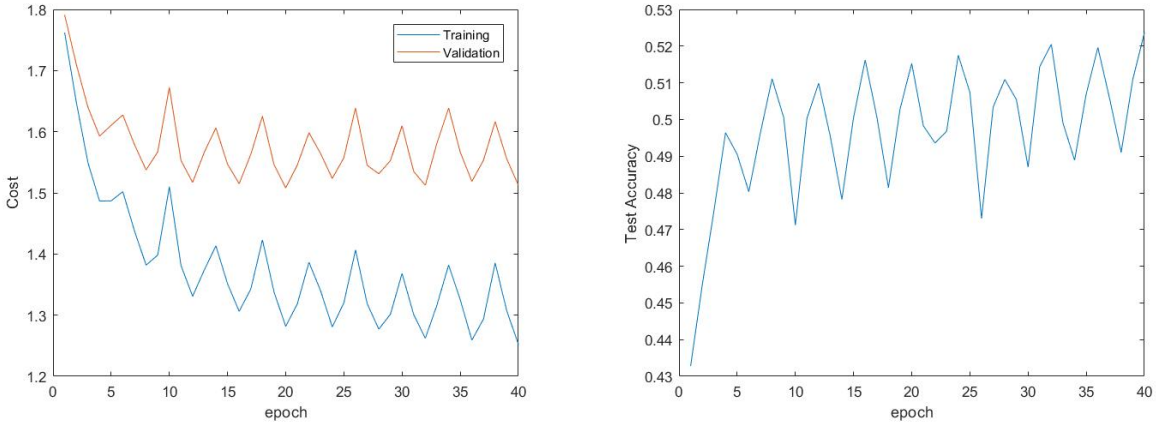
(a) The initial training cost

(b) The initial validation accuracy

Figure 1: Cost and Accuracy as functions of increasing η

From figure 1 it can be seen that the accuracy almost stops increasing and becomes ragged around $\eta = 0.05$, so we set `eta_max = 0.05`, then `eta_min = 0.25 * eta_max` is set empirically. For the final test, 10 cycles' training is employed. The final validation and test accuracy can reach 53.78% and 52.36% respectively. In previous experiment, these two numbers are 53.08% and 52.18% if `eta_min = 1e-5`, `eta_max = 1e-1`, and

2 cycles' training are applied. Our results are slightly better which indicates that the range of learning rate found by this method is reasonable. The final results are shown in figure 2.



(a) The final training and validation cost

(b) The final test accuracy

Figure 2: Final results of the network with newly found η settings

References

[1] Smith, L. N. (2015). Cyclical learning rates for training neural networks. *arXiv:1506.01186 [cs.CV]*.