KTH Royal Institute of Technology

DD2424 Deep Learning in Data Science

Assignment Report 2 bonus

two layer network with multiple outputs bonus part

Written by

Yutong Jiang

yutongj@kth.se

Date: Mar 29, 2022

1 introduction

This bonus assignment could be divided into two parts. For the first part, I try to improve the performance of the network by trying three different methods separately. In the second part, semi-extensive testing is applied.

2 Improvement performance of the network

2.1 Method 1: increase the number of hidden nodes

In method 1, I tried to increase the number of hidden nodes from 50 to 100. In the training process, I load 5 training batched and uses all for training except for 1000 that is used as validation set. The other parameter setting is that lambda is 3.04e-4, n_{batch} is 100, eta_{min} is 1e-5, eta_{max} is 1e-1, n_{cycle} is 3. The final accuracy on test data is 0.5308.

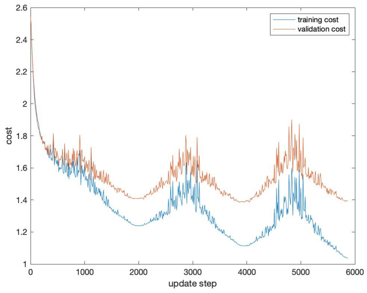

Figure 1: cost curve for 100 hidden nodes

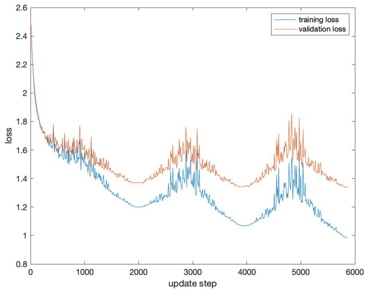

Figure 2: loss curve for 100 hidden nodes

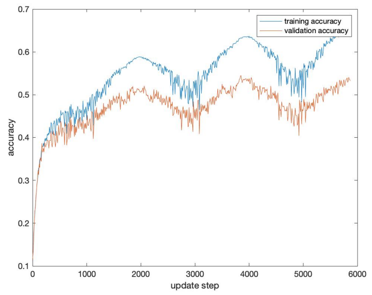

Figure 3: accuracy curve for 100 hidden nodes

However, the number of hidden nodes can not be set too large as it will lead to a more complex network where overfitting happens. Overfitting of the network could lead to a lower accuracy of the test data. Hence, we need to add more regularization and apply dropout method to prevent overfitting.

2.2 Method 2: Apply dropout to the training with increasing hidden nodes

In method 2, I intend to apply dropout to the network which has 500 hidden nodes. And I expect that the accuracy could increase.

The dropout should be applied between the first layer and second layer. The code is shown as below

$$H = max(W_1 * X + b_1) \tag{1}$$

$$U1 = (rand(size(H,1), size(H,2)) < p)/p \tag{2}$$

$$H = H. * U1 \tag{3}$$

In this method, I set p equals to 0.9 and the number of hidden nodes I set is 500, the other parameters stay the same as I did in the Method 1. The final accuracy on test data is 0.5619, which is greater.

2.3 Method 3: apply data augmentation

In method 3, I intend to implement data augmentation and I expect there is an increase in the accuracy of test data.

In the data augmentation part, I flip the images over vertical axis by using the built-in function called *fliplr* in MATLAB and then flatten the flipped images. Hence, the data augmentation is implemented and I get 50000 extra data for training. In the test part, the number of hidden nodes I set is 100 without dropout. The final accuracy I achieved is 0.5370, which is higher than the accuracy in method 1.

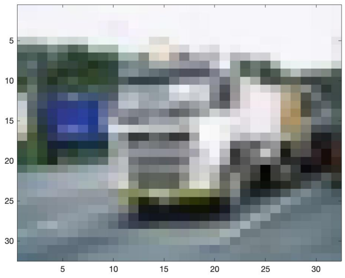

Figure 4: original image of CIFAR 10

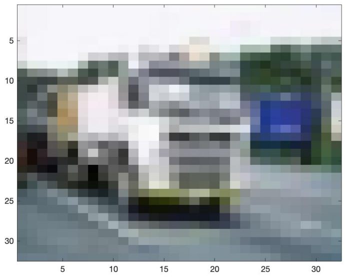

Figure 5: flipped image of CIFAR 10

2.4 conclusion

By applying 3 methods, we could find that there is an improvement on the test accuracy in each method. And the best test accuracy is achieved by method 2, which is 0.5619 by applying dropout with increasing number of hidden nodes to make the network wider.

3 Further improvement

In this part, the target is to see what level of performance I can get with semi-extensive testing.

As the three methods I tried before gives me some positive feedback, I intend to use the combination of 3 methods, which is to train the network with more hidden nodes and dropout by using the augmented data. Besides, I have also tried to adjust some parameters, such as batch size, number of cycles.

The process of finding the best learning rate range is the same as the search I applied in basic part, which is to get random samples from the given range and calculate the accuracy. After that, fine search could be applied. Finally, the range I find for learning rate is from 1e-4 to 1e-1.

Then I changed the n_{cycle} from 3 to 9 and $batch_{size}$ from 200 to 1500 with step 100.

By trying different combination, the best test accuracy is which is 0.5724 given eta_{max} = 1e-1, eta_{min} = 1e-4, number of hidden nodes = 500, $batch_{size}$ = 1000, length of training set = 99000, n_{cycle} is 7.