Report for Assignment 1 Bonus Point

DD2424 Deep Learning in Data Science

Jiaying Yang
950826-9124

1 Exercise 2.1

1. Results

	Training Accuracy	**Validation Accuracy**	**Test Accuracy**
Before Improvement	**41.61%**	**36.6%**	**36.65%**
After Improvement	**41.31%**	**40.40%**	**40.05%**

The gap between the accuracy of training data and test data before improvement is quite big, which means that it is a kind of overfitting. This kind of problem is decreased after my improvement. The test accuracy increased from 36.65% to 40.05%, which is a 9.28% increase with respect to the original accuracy.

2. Improvements chosen

1) ***Use all the available training data for training (all five batches minus a small subset of the training images for a validation set). Decrease the size of the validation set down to ~ 1000.***
All of the 5 data batch are used. The first four data batch and the first 9000 data of the 5th data batch, which in total is 49000 data, is used as training data. The last 1000 data of the 5th data batch is used as validation data.

2) ***Do a grid search to find good values for the amount of regularization and the learning rate.***
I set 3 values of regularization parameter lambda and 3 values of the learning rate eta, and in total there is 9 cases. Then I use two loops to calculate all of the 9 cases (3 by 3), and find out the case that have the best accuracy.

3) ***Play around with decaying the learning rate by a factor ~ :9 after each epoch. Or you can decay the learning rate by a factor of 10 after n epochs.***
I finished this step by decaying the learning rate by a factor of 0.9 for every 10 epochs.

4) ***Implement Xavier initialization and comment if it stabilizes training.***
This is an one layer network with 10 nodes. Thus according to the Xavier Initialization, the weights are initialized by multiplying $\frac{1}{\sqrt{10}}$.

3. Analysis
I think that the 2nd improvement I chosen, which is to do a grid search, helps improve the accuracy best. This is because a grid of parameter settings can be used for parameter optimization, finding out better parameter pair of the learning rate and regularization parameter.

2 Exercise 2.2

In this exercise I change the functions to calculate the cost, gradient, accuracy and the function to realize the mini-batch, in order to train this network in a method of SVM, instead of cross-entropy. I compare the performance of SVM loss and cross-entropy loss by given them same parameter settings, same training data (the 1st batch) and same initialization, whose result is shown as the table below:

lambda	n_epochs	n_batch	eta	Test Accuracy	
				SVM	cross-entropy
0	40	100	0.1	23.82%	26.93%
0	40	100	0.01	30.61%	36.65%
0.1	40	100	0.01	27.00%	33.37%
1	40	100	0.01	20.55%	21.92%
0	40	100	0.001	35.51%	35.65%
0.1	40	100	0.001	34.79%	35.32%

Table 1. Comparison of the test accuracy between SVM and cross entropy in different cases

As is shown in the table above, we can find out that the performance of SVM is always worse than that of the cross-entropy. However when I decrease the value of learning rate eta into 0.001, the performance of SVM improves a lot, though its accuracy is still below than the cross-entropy's. Thus we can arrive at the conclusion that when the speed of convergent is set to a relatively small value, using the SVM-multi-class loss to train the network is more likely to get a better result.