DD2424 Deep Learning in Data Science
Assignment 1
Bonus

Rui Shi
srui@kth.se

April 6, 2022

1 Introduction

This is the bonus section of assignment 1. In this section, I tried 4 ways to improve performance of the network: Use all the available training data for training and decrease the size of the validation set; decay the learning rate; Xavier initialisation; increase number of epochs with smaller learning rate.

2 Improve performance of the network

2.1 Use all the available training data for training and decrease the size of the validation set

For the basic part of the experiment, 1 batch was used for training(n=10000) and 1 batch was used for validation. In this section, 4 batches and part of the fifth batch were used for training(n=49000) and the size of the validation set is decreased to 1000. The other parameters are set as: lambda = 1, n_epochs = 40, n_batch = 100, eta = .001

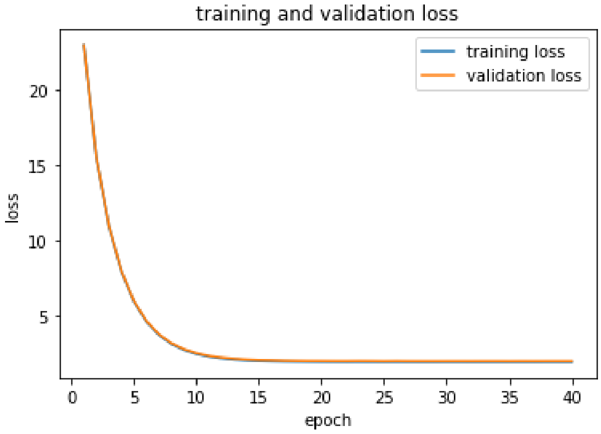

Figure 1: Graph of the loss on the training data and validation data

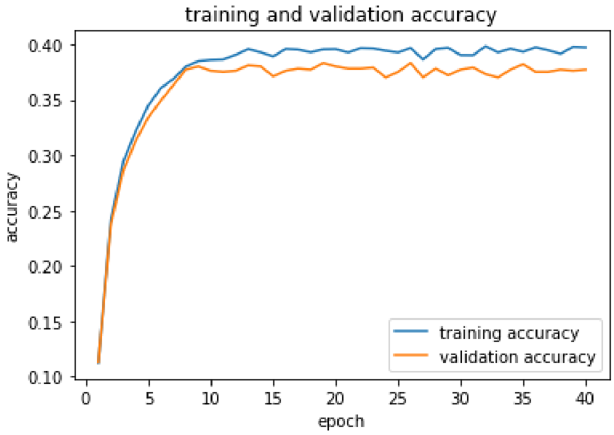

Figure 2: Graph of the accuracy on the training data and validation data

	Basic	Improvement
Train cost	1.9251959196467456	1.9055450510861416
Train accuracy	0.3961	0.3971
Validation cost	1.958542459241439	1.9599314449835665
Validation accuracy	0.3628	0.377
Test cost	1.9378516126671532	1.937421751188373
Test accuracy	0.3628	0.377

2.2 Step decay

In this section, the decay rate is set to 0.95. The other parameters are set as: lambda = 1, n_epochs = 40, n_batch = 100, eta = .001.

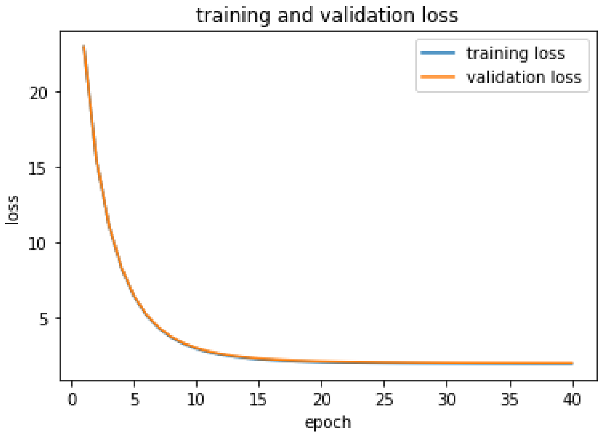

Figure 3: Graph of the loss on the training data and validation data

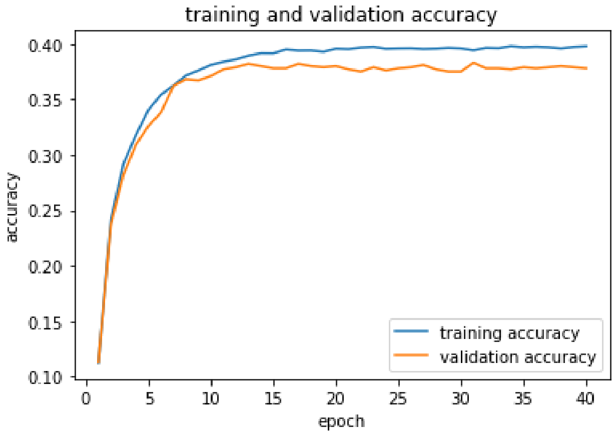

Figure 4: Graph of the accuracy on the training data and validation data

	Basic	Improvement
Train cost	1.9251959196467456	1.9238062165915268
Train accuracy	0.3961	0.3978
Validation cost	1.958542459241439	1.977983906519832
Validation accuracy	0.3628	0.378
Test cost	1.9378516126671532	1.9541827684447282
Test accuracy	0.3628	0.378

2.3 Xavier initialisation

In this section, the decay rate is set to 0.95. The other parameters are set as: lambda = 1, n_epochs = 40, n_batch = 100, eta = .001.

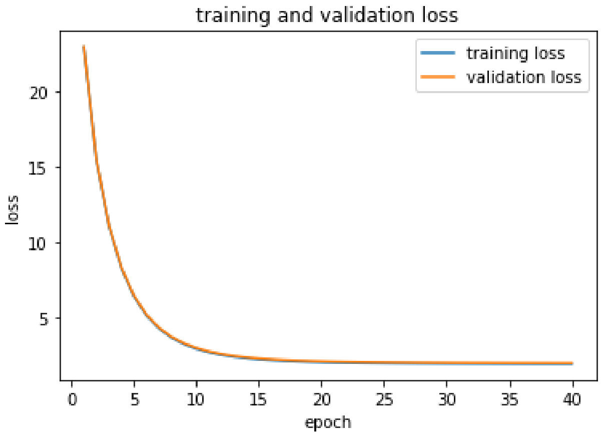

Figure 5: Graph of the loss on the training data and validation data

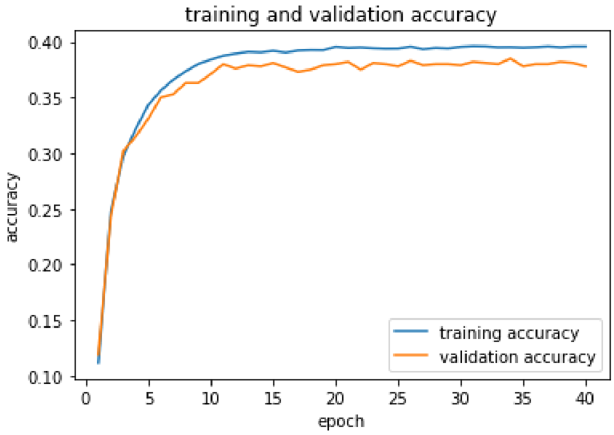

Figure 6: Graph of the accuracy on the training data and validation data

	Basic	Improvement
Train cost	1.9203357071130245	1.980214131075162
Train accuracy	0.3961	0.3958
Validation cost	1.958542459241439	1.977983906519832
Validation accuracy	0.3628	0.378
Test cost	1.9378516126671532	1.9568649610520508
Test accuracy	0.3628	0.378

2.4 Increase number of epochs and decrease learning rate

Finally, I increase the number of epochs to 400 and decrease the learning rate to 0.0003, wish to have a more accurate result.

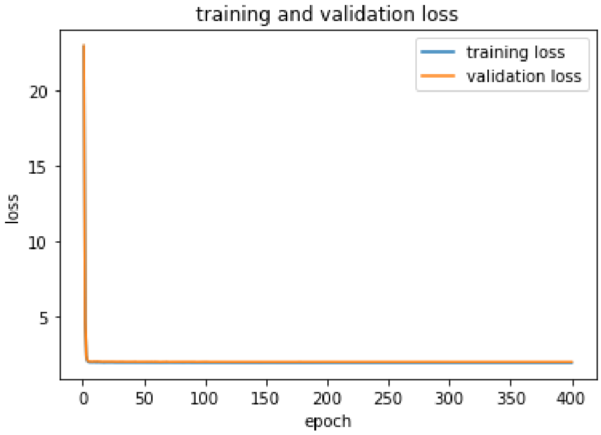

Figure 7: Graph of the loss on the training data and validation data

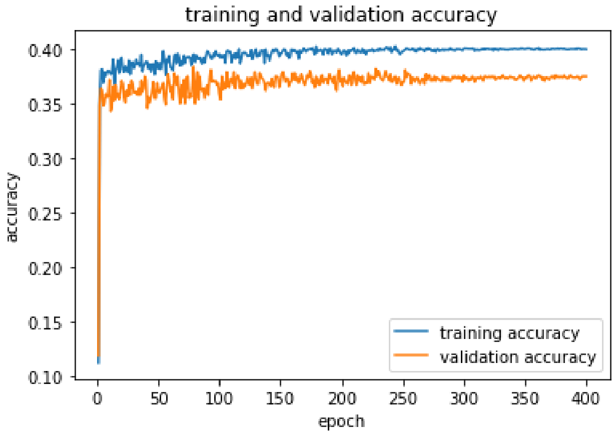

Figure 8: Graph of the accuracy on the training data and validation data

	Basic	Improvement
Train cost	1.9203357071130245	1.8990442284559945
Train accuracy	0.3961	0.3999
Validation cost	1.958542459241439	1.963335654702013
Validation accuracy	0.3628	0.375
Test cost	1.9378516126671532	1.9315630045686623
Test accuracy	0.3628	0.382

2.5 Combination and conclusion

After times of experiments, I combine these 4 methods and finally get the best model. The parameters are set as below:

$$lambda = 1, n\_epochs = 400, n\_batch = 100, eta = .0005, decay\_factor = 0.99 \tag{1}$$

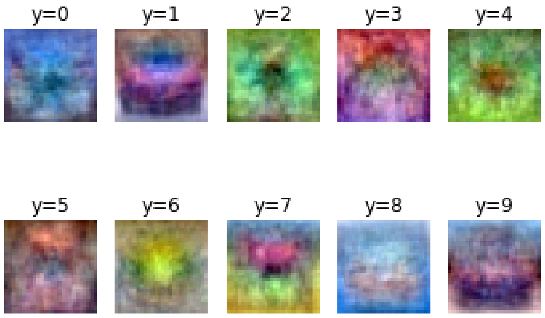

Figure 9: The learnt weight matrix

And the table above shows the improvement, all the performances increase. But it is noticed that the improvements are slight, maybe a two layer classification, or CNN, could cause qualitatively improvements. For my experiments, use all the available training data for training and decrease the size of the validation set

3 Train network - multiple binary cross-entropy losses

3.1 Please write down the expression for $\partial l_{multiplebce}/\partial s$

$$\frac{\partial l_{multiplebce}}{\partial s} = \frac{\partial l_{multiplebce}}{\partial p} * \frac{\partial p}{\partial s} = -(\frac{y}{p} + \frac{y-1}{1-p}) * p * (1-p) = p - y \tag{2}$$

As the equation above, we could find that we do not need to change the code. We just need to replace the softmax operation by sigmoid operation.

3.2 Report the final test accuracy

Train accuracy	Validation accuracy	Test accuracy	
Cross-entropy	0.4008	0.3629	0.3707
Multiple Binary	0.5059	0.372	0.3786

From the accuracy on test data, we find that there is no significant improvement. But we can see that the accuracy on train data, multiple binary cross-entropy improves a lot, which means that more over-fitting occurs with multiple binary cross-entropy.

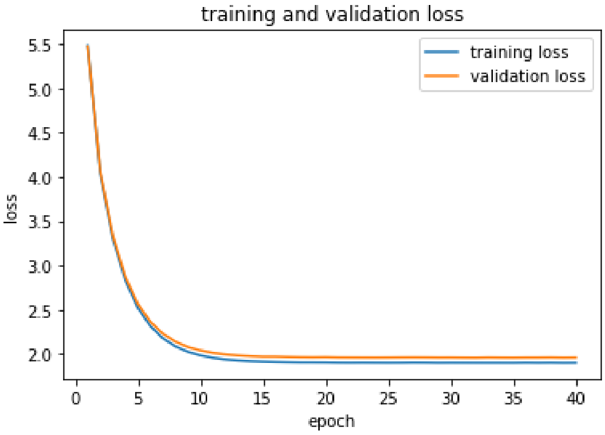

Figure 10: Loss of multiple binary cross-entropy

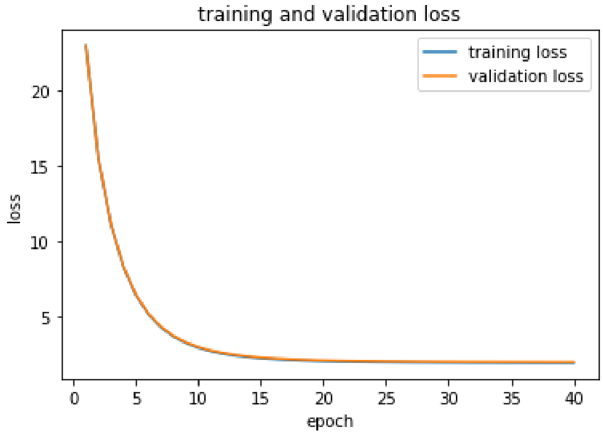

Figure 11: Loss of ca

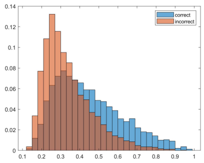

Figure 12: Histogram for softmax + cross-entropy

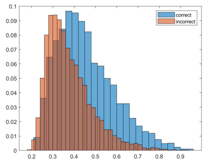

Figure 13: Histogram for sigmoid + multiple binary

From the histograms, we can see that the distribution of binary loss is close to 0, especially for the examples correctly. Because in multiple binary, every element make contributions to the classification.