DD2424 Deep Learning in Data Science

Assignment 1 - Bonus Points

Ramona Häuselmann

April 3, 2021

1 Exercise 2.1

Starting from the results of Experiment 4 of Assignment 1 the following approaches to improve the network performance were tested:

(i) use all available training data

(ii) train for a longer time

The results after Experiment 4 of Assignment 1 (lambda=0, n_epochs=40, n_batch=100, eta=0.1) were as follows:
training loss: 1.899
validation loss: 1.958
accuracy: 37.38%

1.1 Results Improvement (i)

final training loss 1.920
final validation loss 1.935
final accuracy 0.3786

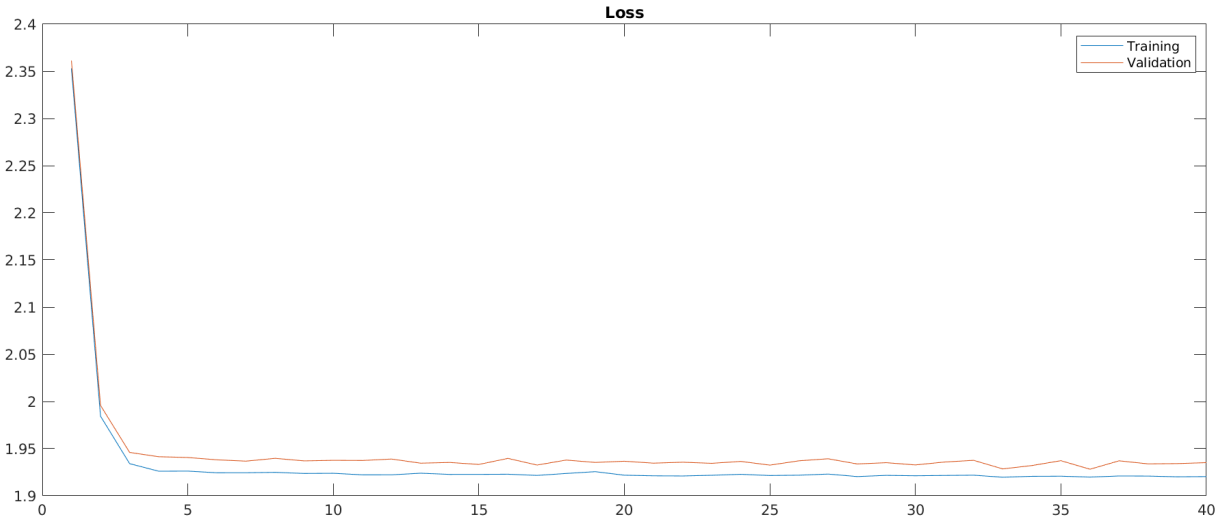

Figure 1: Improvement (i) Loss (lambda=1, n_epochs=40, n_batch=100, eta=0.001)

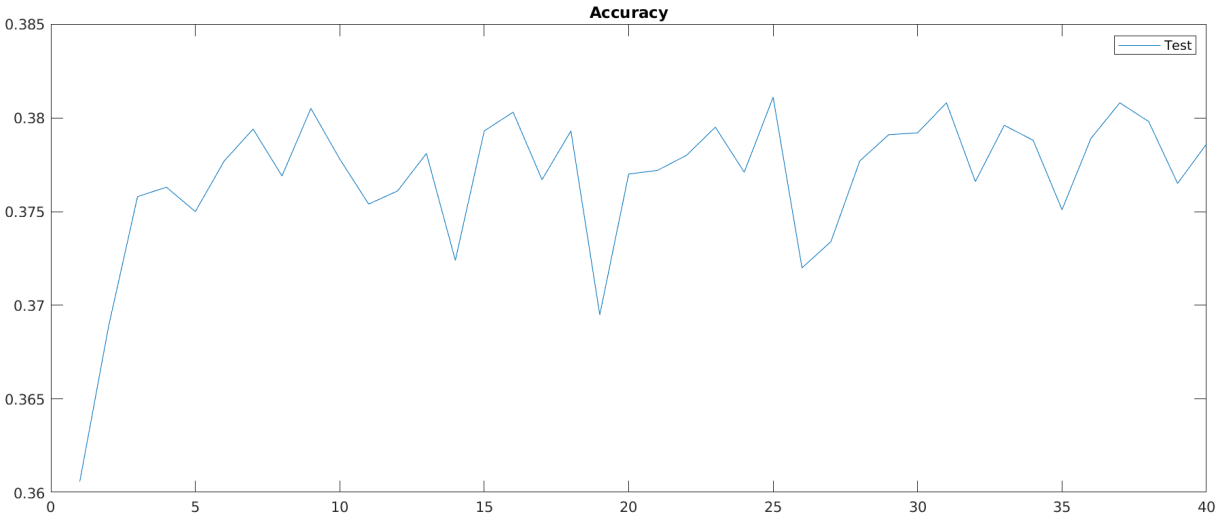

Figure 2: Improvement (i) Accuracy (lambda=1, n_epochs=40, n_batch=100, eta=0.001)

Figure 3: Improvement (i) Weights (lambda=1, n_epochs=40, n_batch=100, eta=0.001)

1.2 Results Improvement (ii)

In this experiment I trained on all available data as in experiment 1 and used the same parameters. I trained the network for 2000 epochs and every 100 epochs I stored a snapshot of the diagrams for loss, accuracy and weights. Also I logged the values of each epoch to a file (result_pics/train_longer/values.csv). After each epoch I compared the training loss to the validation loss and set a threshold (0.5) to detect when the network begins to overfit, but in my runs the values stayed very close to each other and never exceeded that threshold. The maximum absolute difference over all 2000 epochs was 0.01996.
As a result we can observe that training longer does not improve the result that much, but we also see that no overfitting seems to occur. We see that during the first 100 epochs the accuracy increases slightly but after that it stays more or less the same.

epoch	training loss	validation loss	accuracy
1	2.35312	2.36131	36.06%
10	1.92387	1.9376	37.78%
50	1.92102	1.93453	37.55%
100	1.92044	1.93162	37.65%
200	1.91963	1.92828	37.72%
300	1.92069	1.93098	37.64%
400	1.92159	1.93381	37.42%
500	1.92045	1.92874	37.89%
600	1.9197	1.93142	37.90%
700	1.91865	1.93335	38.25%
800	1.91992	1.93714	38.37%
900	1.92071	1.93663	38.03%
1000	1.92166	1.92969	37.43%

Table 1: Summary of longer training

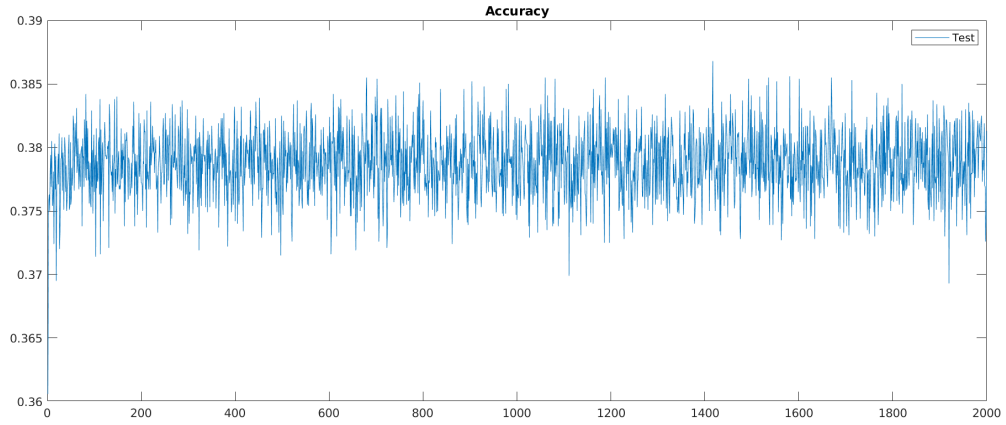

Figure 4: Improvement (ii) accuracy after 2000 epochs

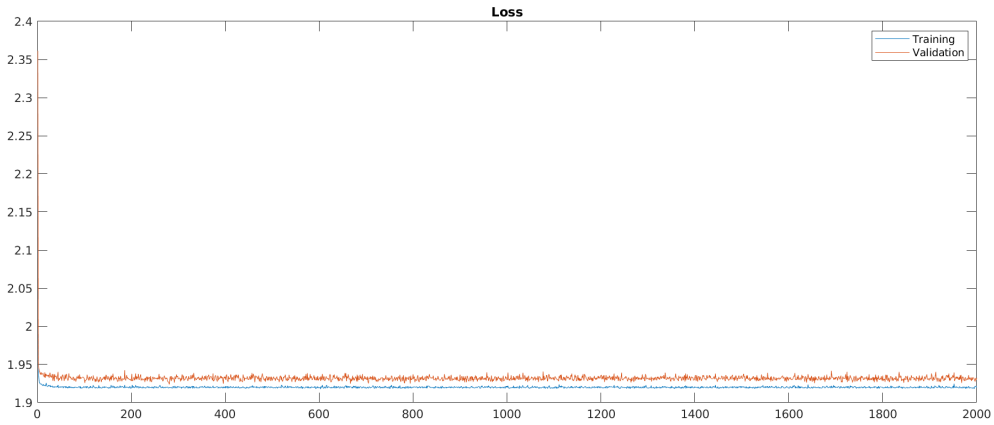

Figure 5: Improvement (ii) Loss after 2000 epochs

Figure 6: Improvement (ii) weights after 2000 epochs

1.3 Experiment 1 diagrams

As seen in the diagrams with the parameters of this run the network is not really able to learn. The loss and accuracy look very random. This results in a very low accuracy (27.84%). The weight matrices look pretty random as well.

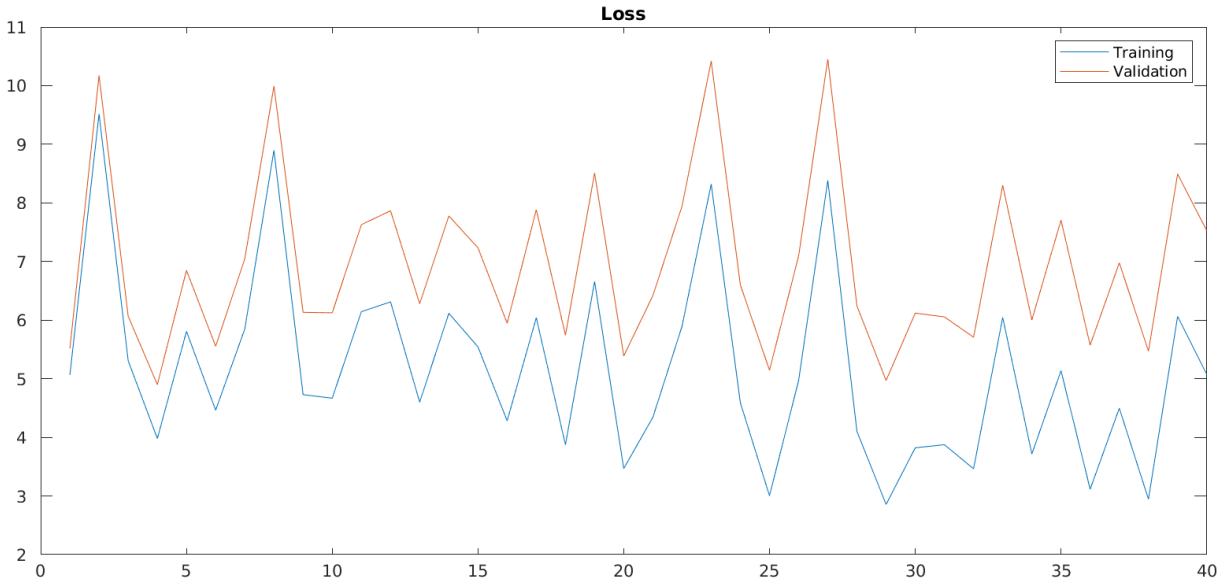

Figure 7: Experiment 1 Loss (lambda=0, n_epochs=40, n_batch=100, eta=0.1)

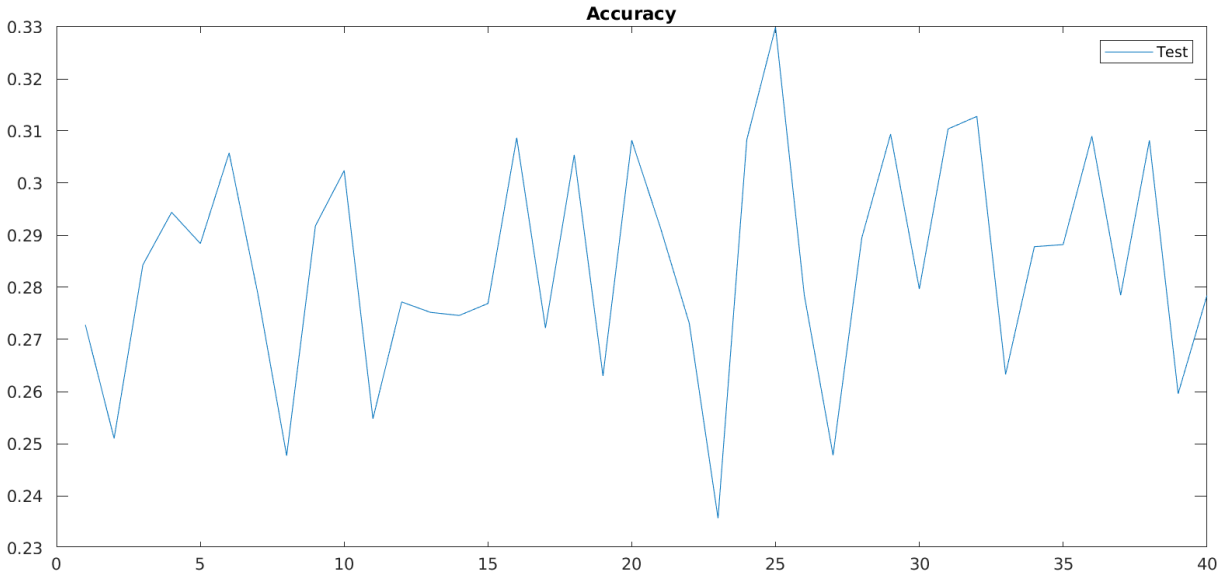

Figure 8: Experiment 1 Accuracy (lambda=0, n_epochs=40, n_batch=100, eta=0.1)

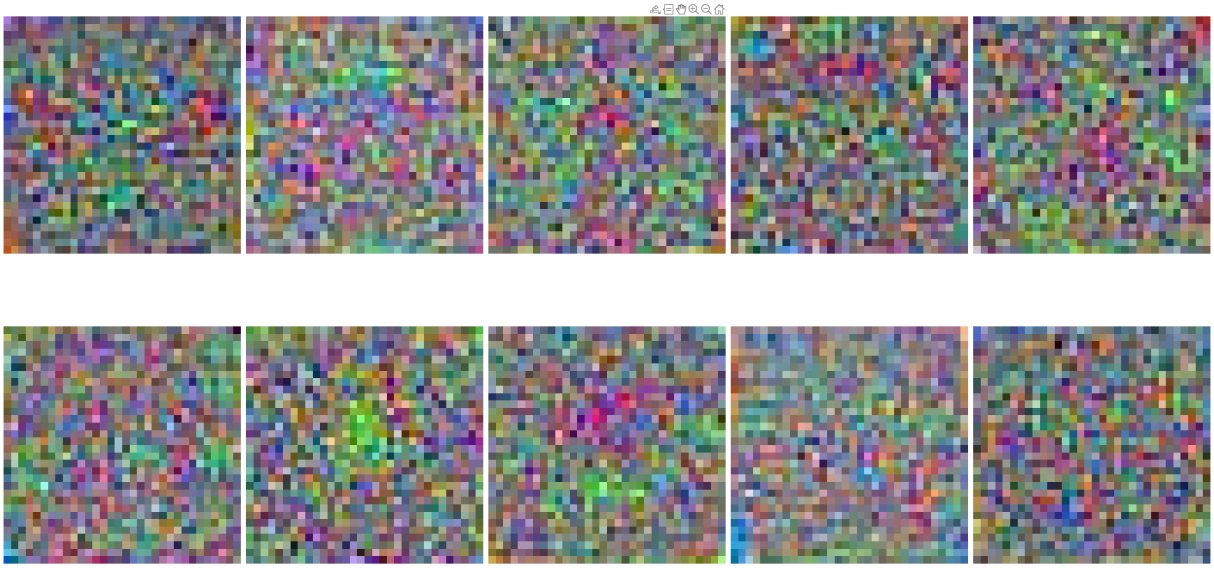

Figure 9: Experiment 1 Weights (lambda=0, n_epochs=40, n_batch=100, eta=0.1)

1.4 Experiment 2 diagrams

Compared to the first experiment in this run we use a much smaller learning rate. In this run it looks like the network learns better and it results in a higher accuracy (39.08%). If we look at the loss graphs we can see that there is a big difference between the training loss and validation loss. This could be an indicator for overfitting. The weight matrices look less random.

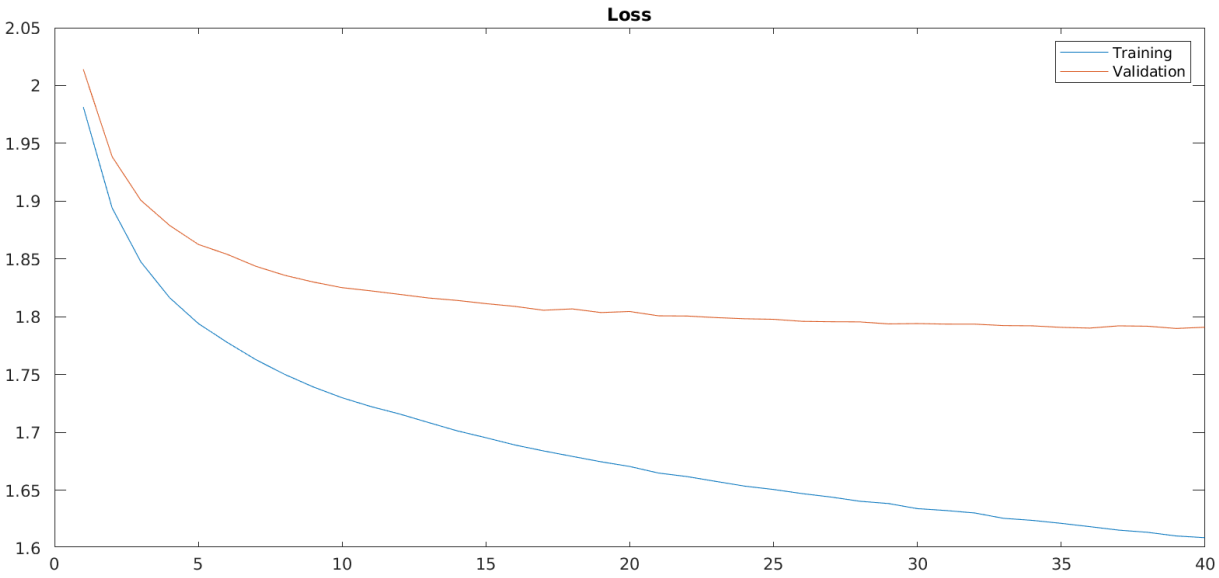

Figure 10: Experiment 2 Loss (lambda=0, n_epochs=40, n_batch=100, eta=0.001)

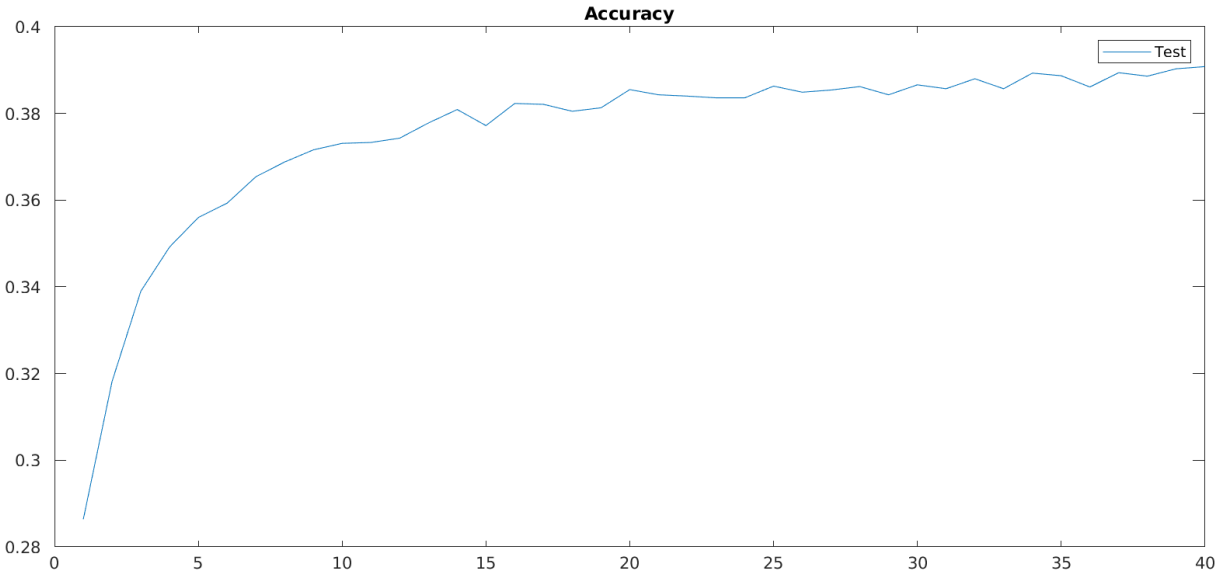

Figure 11: Experiment 2 Accuracy (lambda=0, n_epochs=40, n_batch=100, eta=0.001)

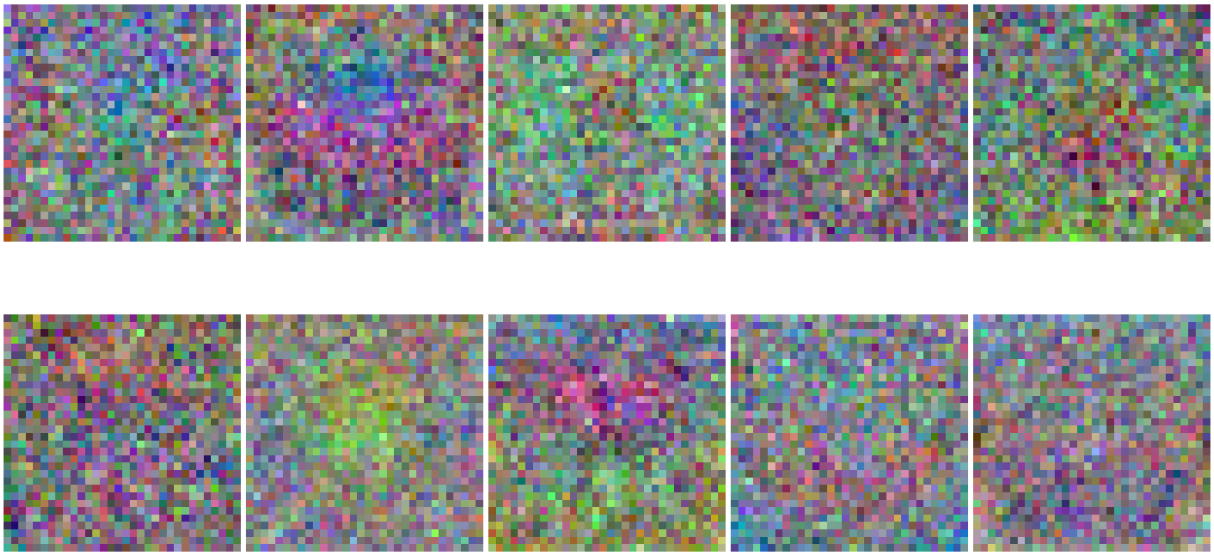

Figure 12: Experiment 2 Weights (lambda=0, n_epochs=40, n_batch=100, eta=0.001)

1.5 Experiment 3 diagrams

In this experiment we use a small regularization term. This results in a smaller training loss compared to the previous run. But we have still a big difference between training and validation which could mean that we overfit. We start to see some more distinct patterns in the weight matrices.

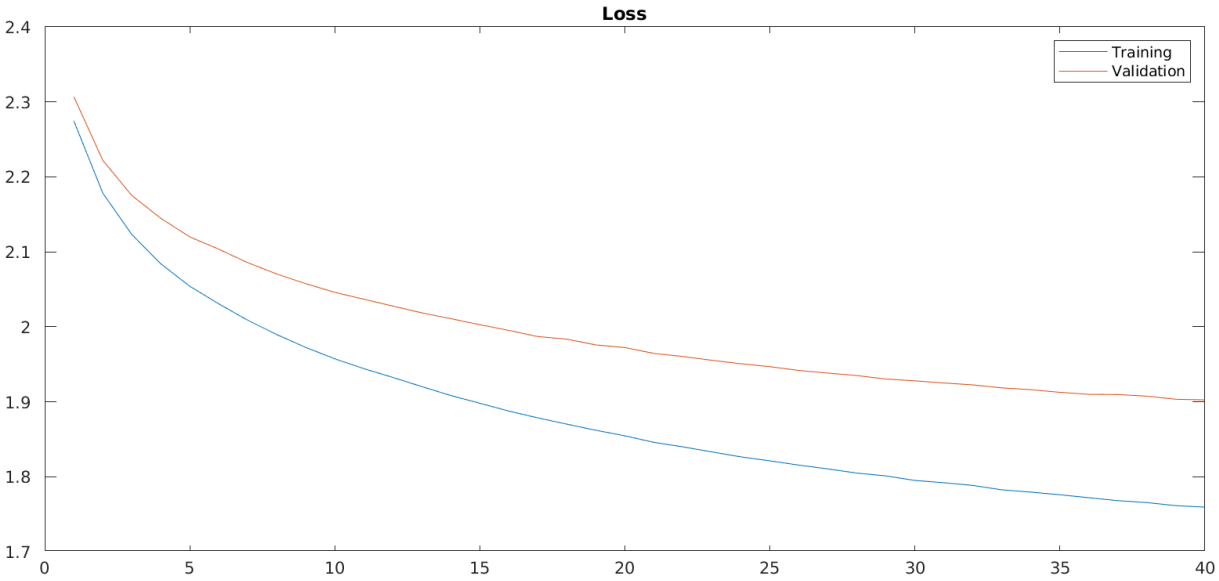

Figure 13: Experiment 3 Loss (lambda=0.1, n_epochs=40, n_batch=100, eta=0.001)

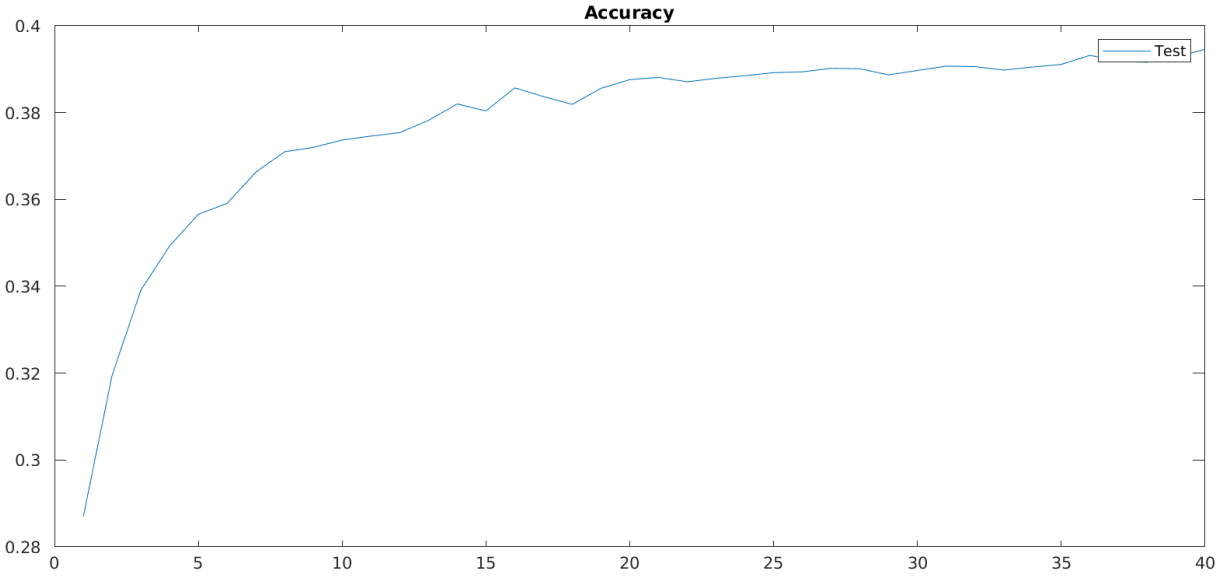

Figure 14: Experiment 3 Accuracy (lambda=0.1, n_epochs=40, n_batch=100, eta=0.001)

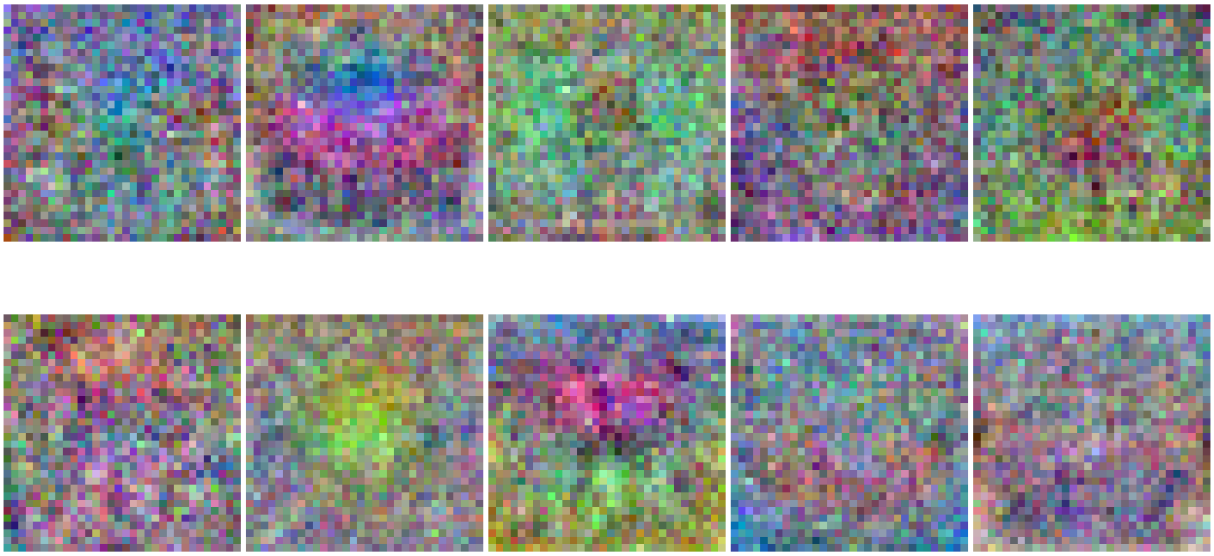

Figure 15: Experiment 3 Weights (lambda=0.1, n_epochs=40, n_batch=100, eta=0.001)

1.6 Experiment 4 diagrams

In this experiment we use an even bigger regularization term. That results in a slightly reduced accuracy but also the training and validation loss are close. That could mean that we are not overfitting anymore. In the weight matrix we see that we start to learn some patterns that looks similar to the images of the corresponding classes. Maybe with more training these patterns become even more visible.

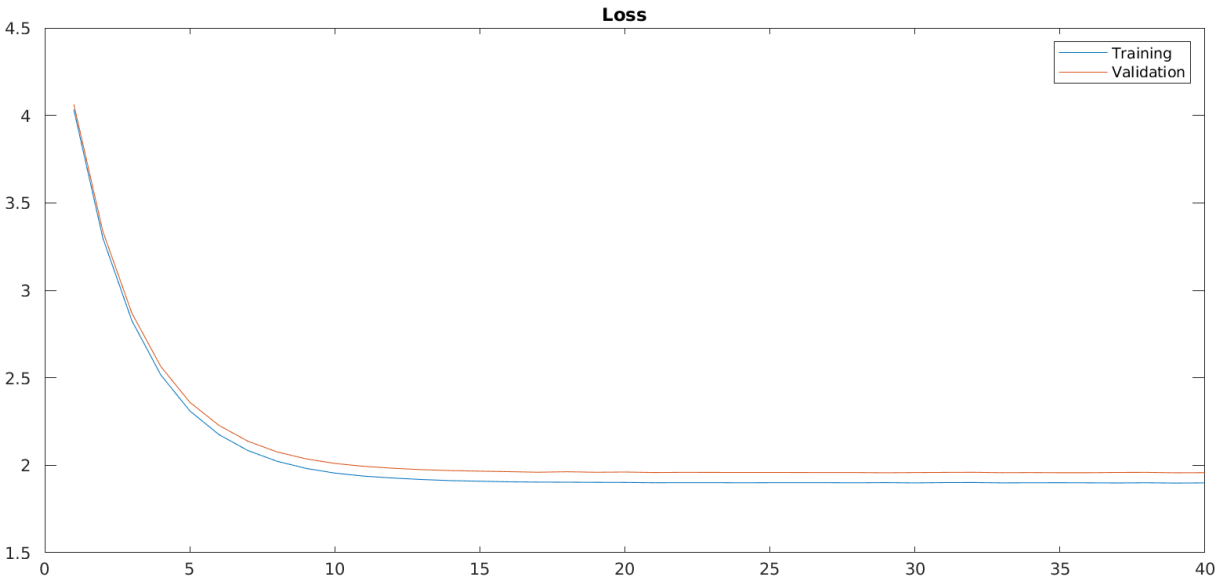

Figure 16: Experiment 4 Loss (lambda=1, n_epochs=40, n_batch=100, eta=0.001)

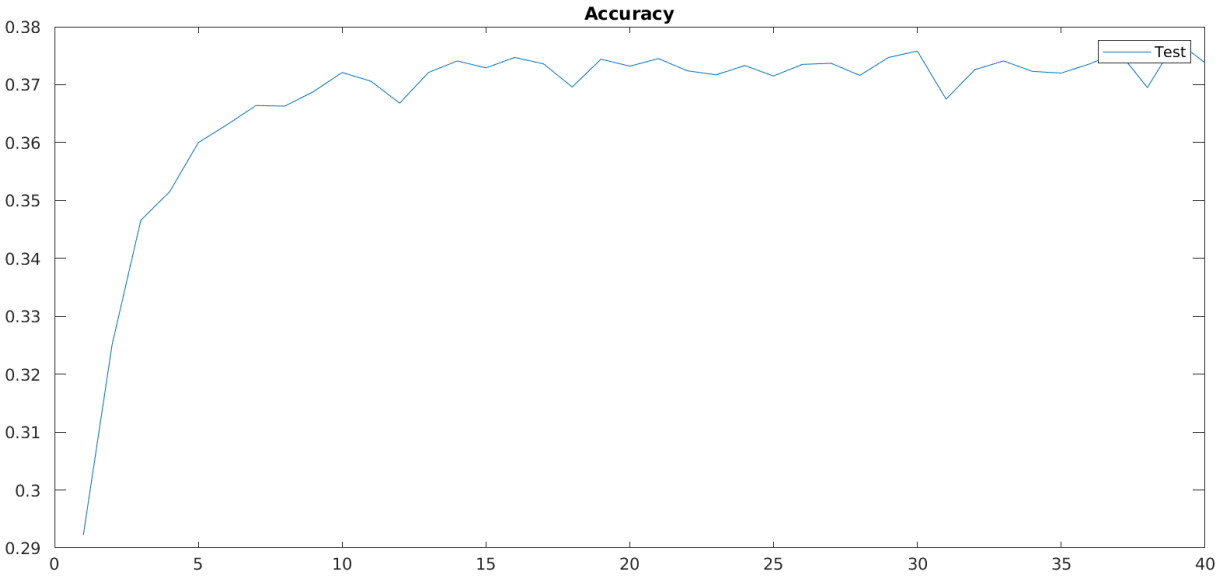

Figure 17: Experiment 4 Accuracy (lambda=1, n_epochs=40, n_batch=100, eta=0.001)

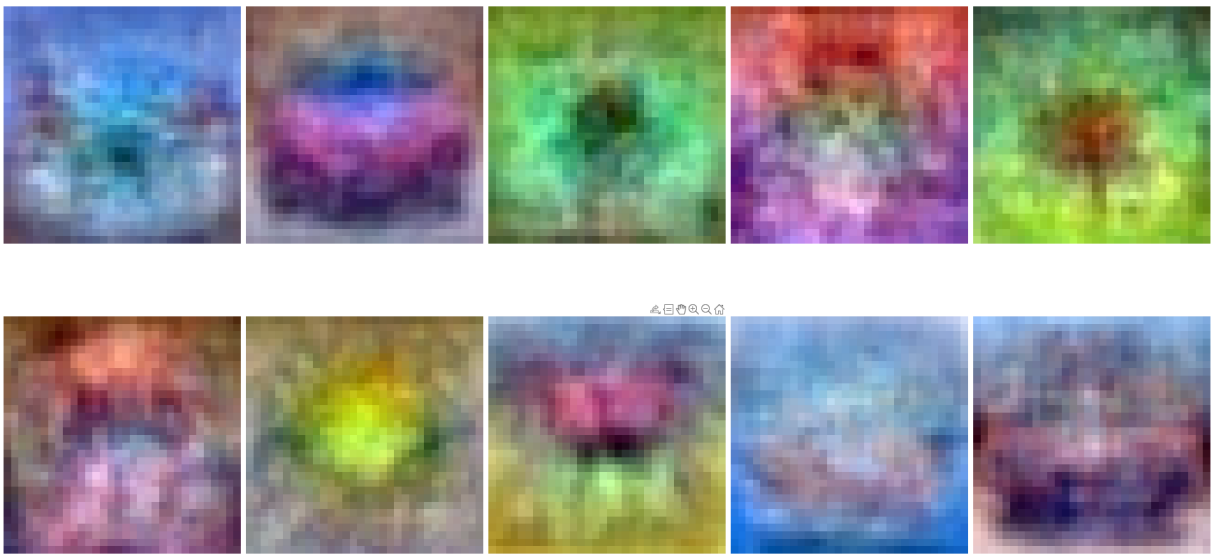

Figure 18: Experiment 4 Weights (lambda=1, n_epochs=40, n_batch=100, eta=0.001)

2 Conclusion

Increasing the amount of regularization reduces the accuracy but also prevents the network from overfitting. The choice of the correct learning rate is very important for the networks ability to learn, as we can see when comparing Experiment 1 and Experiment 2. The learning rate has a huge impact on the accuracy.