DD2424 Deep Learning in Data Science

Assignment 1

Ramona Häuselmann

April 2, 2021

1 Gradient Computation

I managed to successfully write the functions to correctly compute the gradient analytically. I tested my implementation by computing the gradients using both ComputeGradsNum.m and ComputeGradsNumSlow.m. Then I compared my results with the numerical approaches by calculating the absolute difference of each gradient element as in equation (1) and (2). Then I checked those against a threshold (1e-6). I also used the other formula that was given in the assignment instructions to compare the error (equation (3) and (4)). Depending on the seed sometimes I got an error that was larger than 1e-6 (e.g. 3e-6) but this error is still small enough to conclude that my implementation works.

$$diff\_W = abs(ngrad\_W - grad\_W) \tag{1}$$

$$diff\_b = abs(ngrad\_b - grad\_b) \tag{2}$$

$$diff\_W = abs(ngrad\_W - grad\_W)./max(eps, abs(grad\_W) + abs(ngrad\_W)) \tag{3}$$

$$diff\_b = abs(ngrad\_b - grad\_b)./max(eps, abs(grad\_b) + abs(ngrad\_b)) \tag{4}$$

2 Results

	Experiment 1	**Experiment 2**	**Experiment 3**	**Experiment 4**
seed	400	400	400	400
lambda	0	0	0.1	1
n_batch	100	100	100	100
eta	0.1	0.001	0.001	0.001
n_epochs	40	40	40	40

Table 1: Experiment parameters

	Experiment 1	**Experiment 2**	**Experiment 3**	**Experiment 4**
Training	5.074	1.609	0.3908	1.899
Validation	7.526	1.791	1.903	1.958

Table 2: Summary of final loss

	Experiment 1	**Experiment 2**	**Experiment 3**	**Experiment 4**
Accuracy	27.84%	39.08%	39.46%	37.38%

Table 3: Summary of final accuracy

2.1 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.

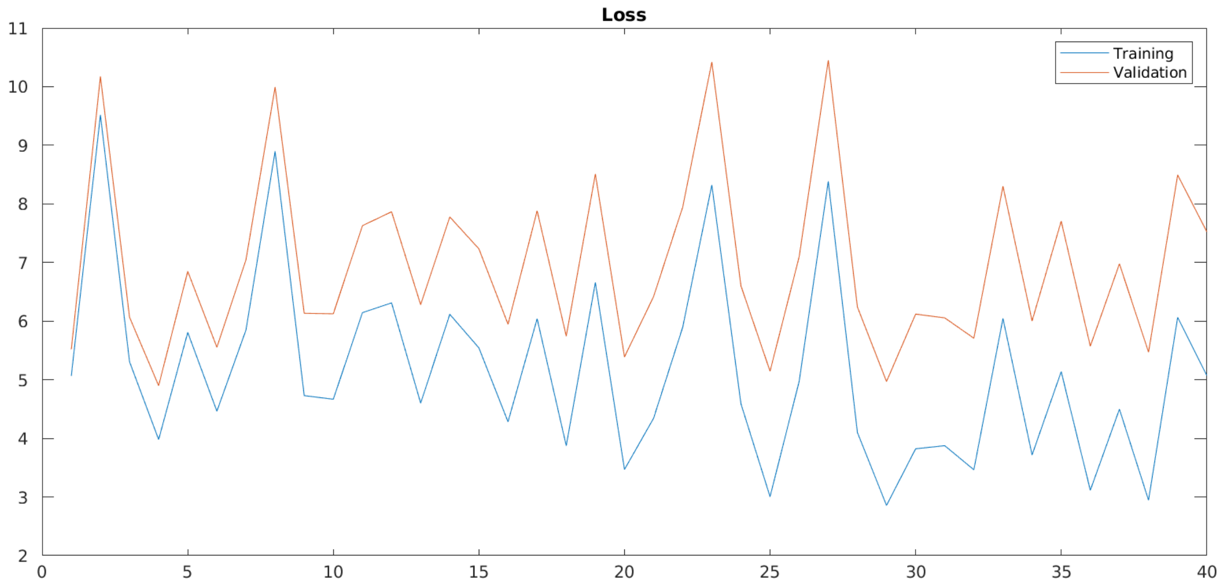

Figure 1: Experiment 1 Loss (lambda=0, n_epochs=40, n_batch=100, eta=0.1)

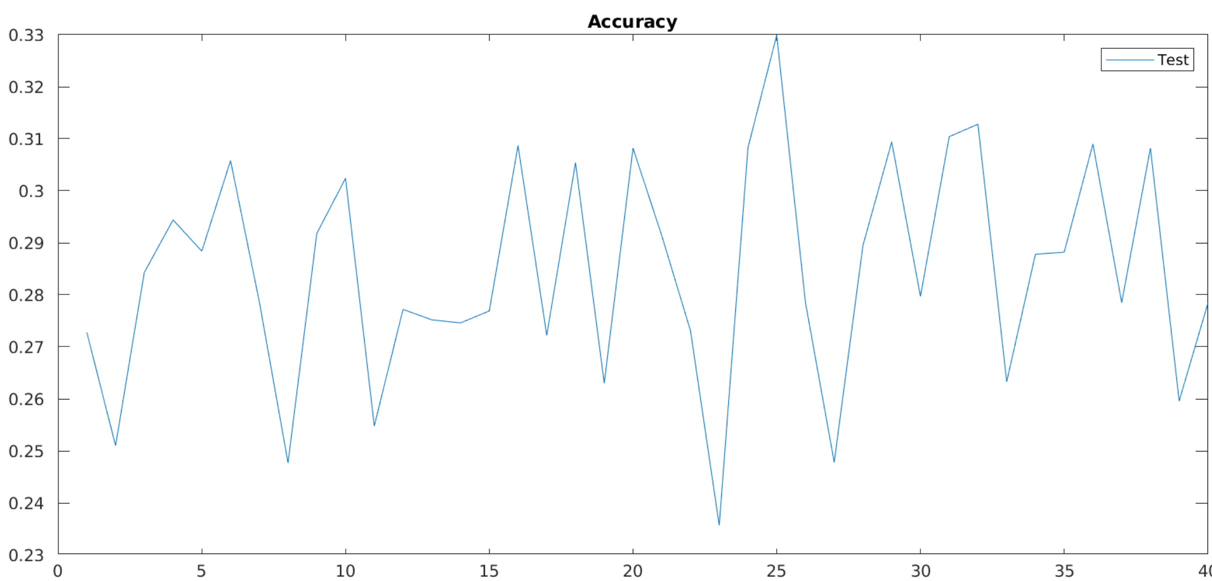

Figure 2: Experiment 1 Accuracy (lambda=0, n_epochs=40, n_batch=100, eta=0.1)

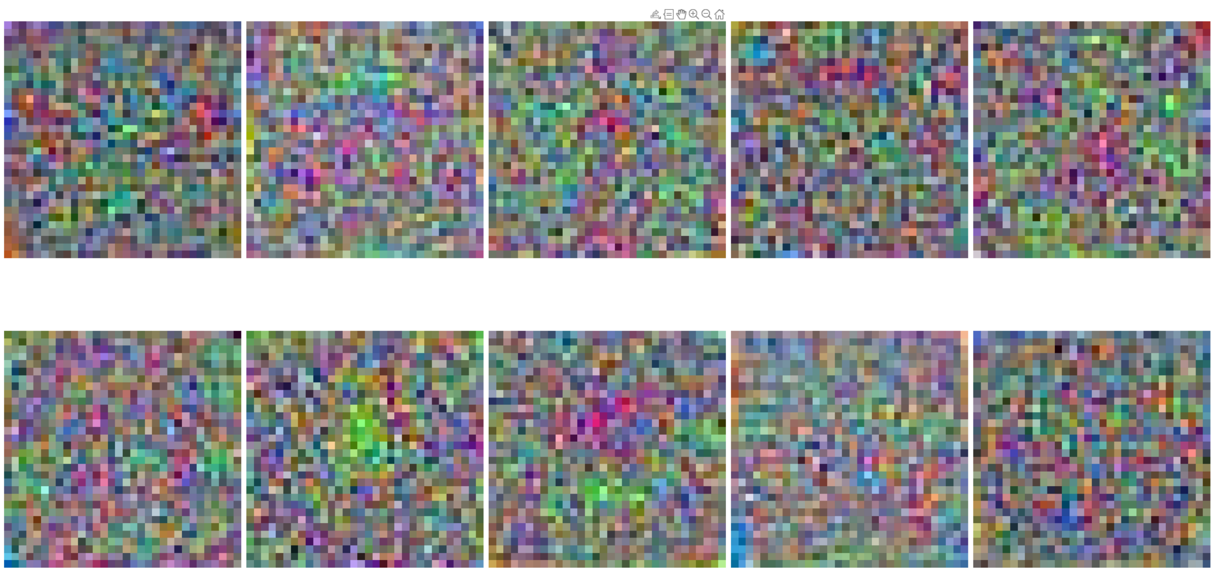

Figure 3: Experiment 1 Weights (lambda=0, n_epochs=40, n_batch=100, eta=0.1)

2.2 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.

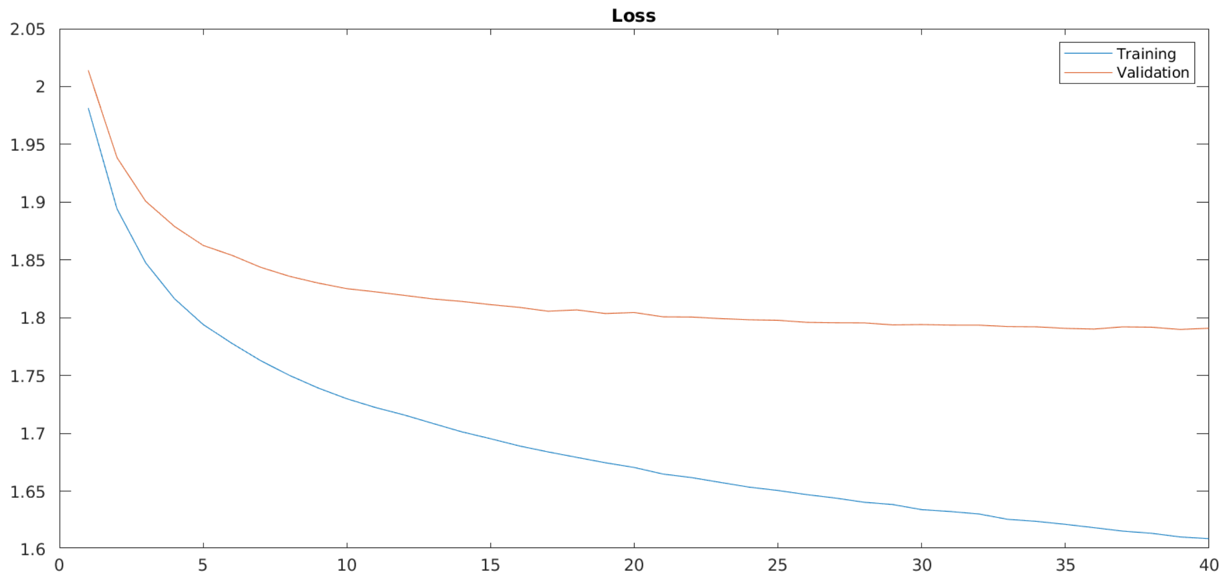

Figure 4: Experiment 2 Loss (lambda=0, n_epochs=40, n_batch=100, eta=0.001)

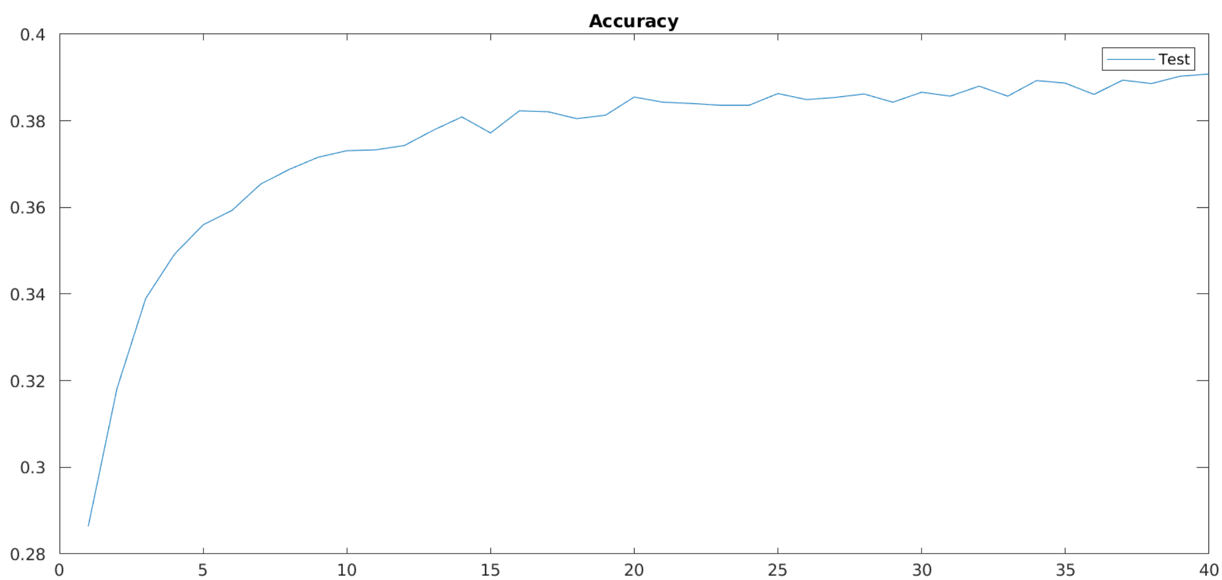

Figure 5: Experiment 2 Accuracy (lambda=0, n_epochs=40, n_batch=100, eta=0.001)

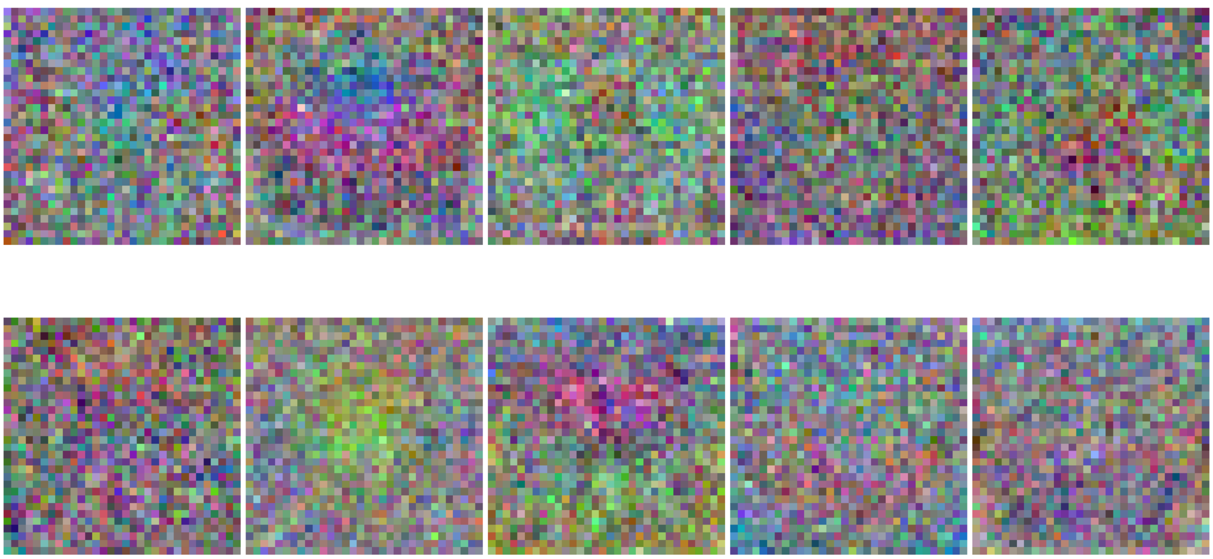

Figure 6: Experiment 2 Weights (lambda=0, n_epochs=40, n_batch=100, eta=0.001)

2.3 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.

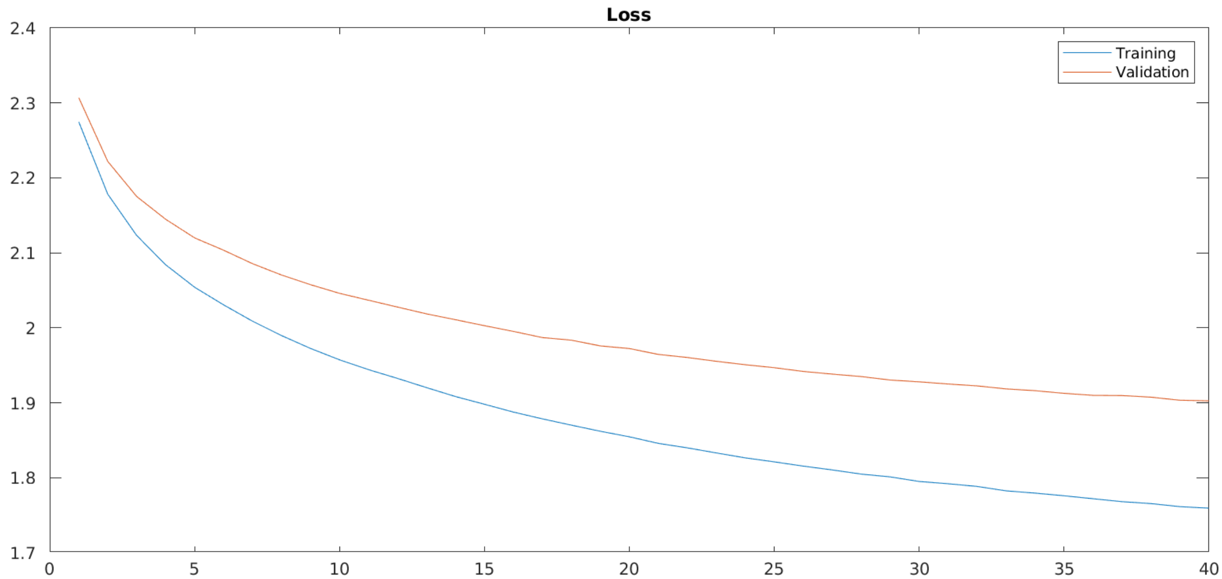

Figure 7: Experiment 3 Loss (lambda=0.1, n_epochs=40, n_batch=100, eta=0.001)

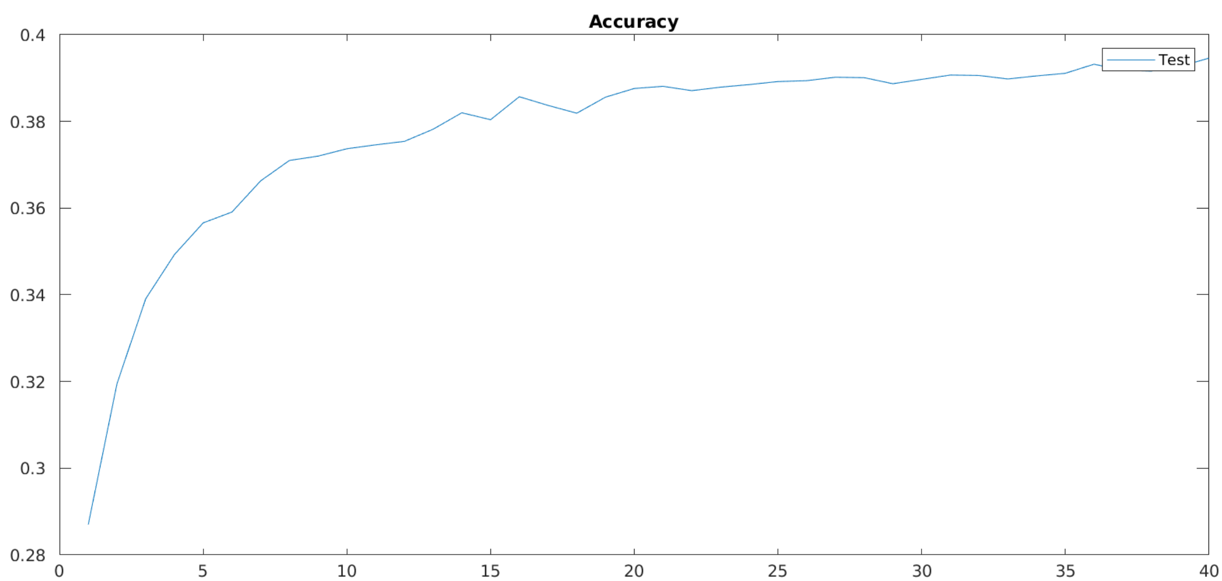

Figure 8: Experiment 3 Accuracy (lambda=0.1, n_epochs=40, n_batch=100, eta=0.001)

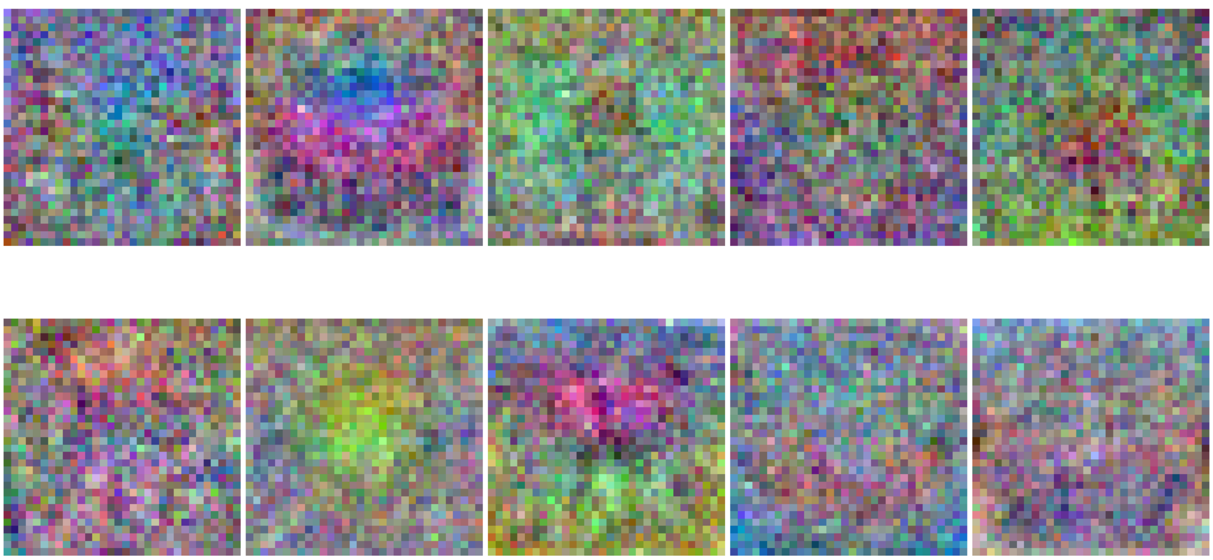

Figure 9: Experiment 3 Weights (lambda=0.1, n_epochs=40, n_batch=100, eta=0.001)

2.4 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.

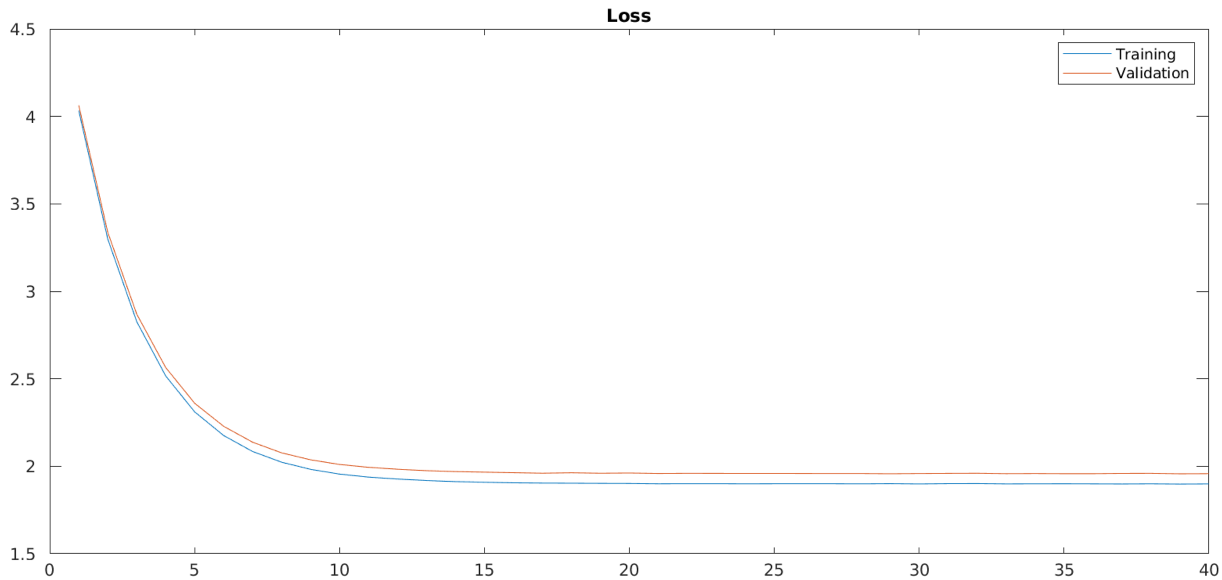

Figure 10: Experiment 4 Loss (lambda=1, n_epochs=40, n_batch=100, eta=0.001)

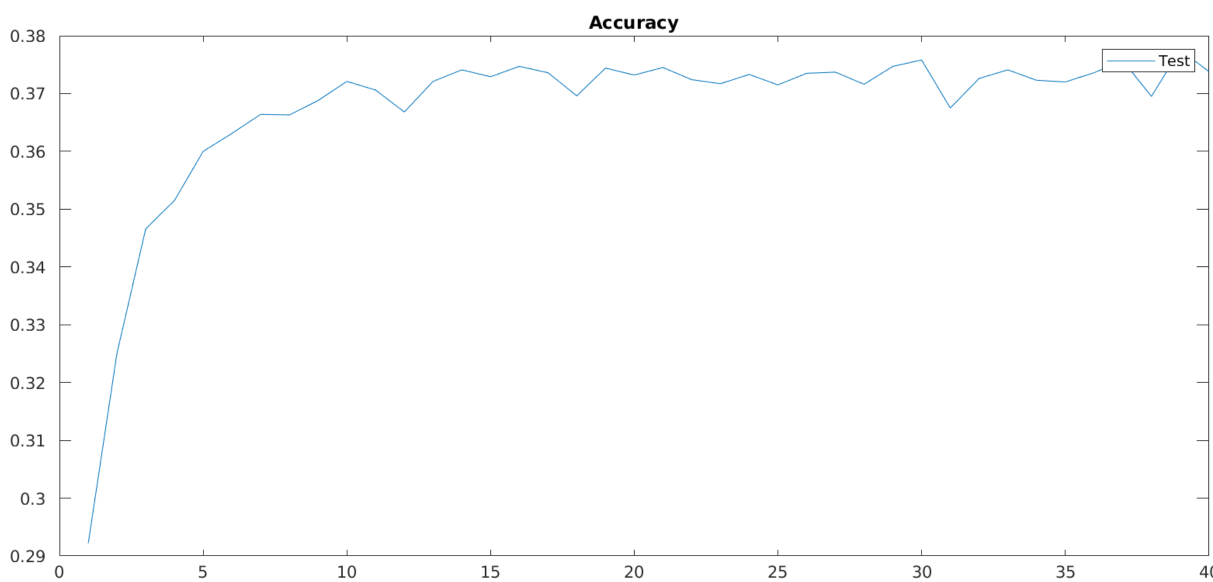

Figure 11: Experiment 4 Accuracy (lambda=1, n_epochs=40, n_batch=100, eta=0.001)

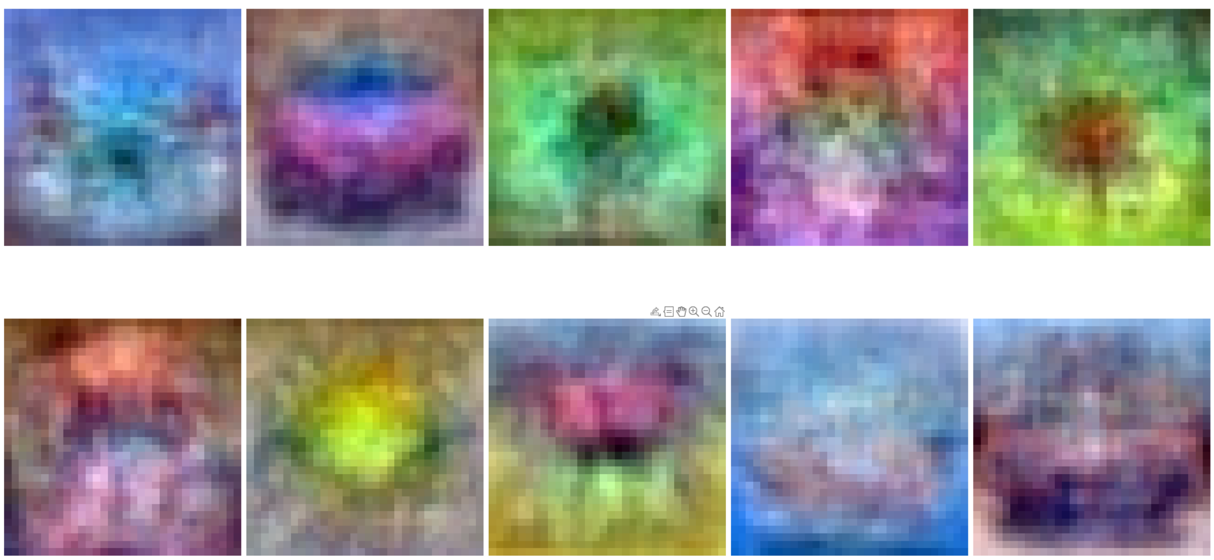

Figure 12: Experiment 4 Weights (lambda=1, n_epochs=40, n_batch=100, eta=0.001)

3 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.