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

(iii) shuffle the order of the training set at the beginning of every epoch

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

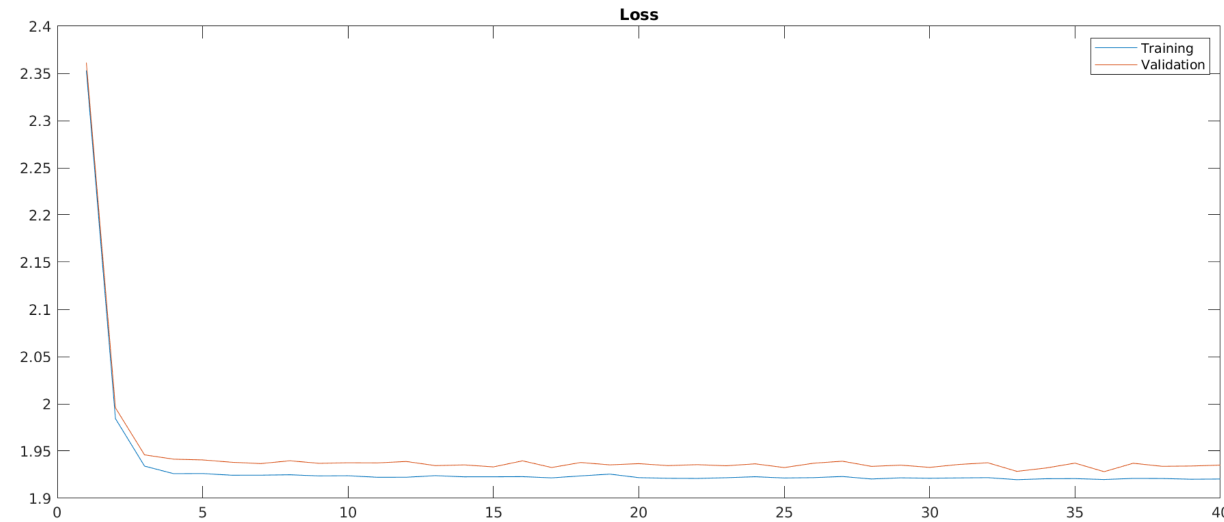


Figure 1: Improvement (i) Loss (lambda=1, n_epochs=40, n_batch=100, eta=0.001)

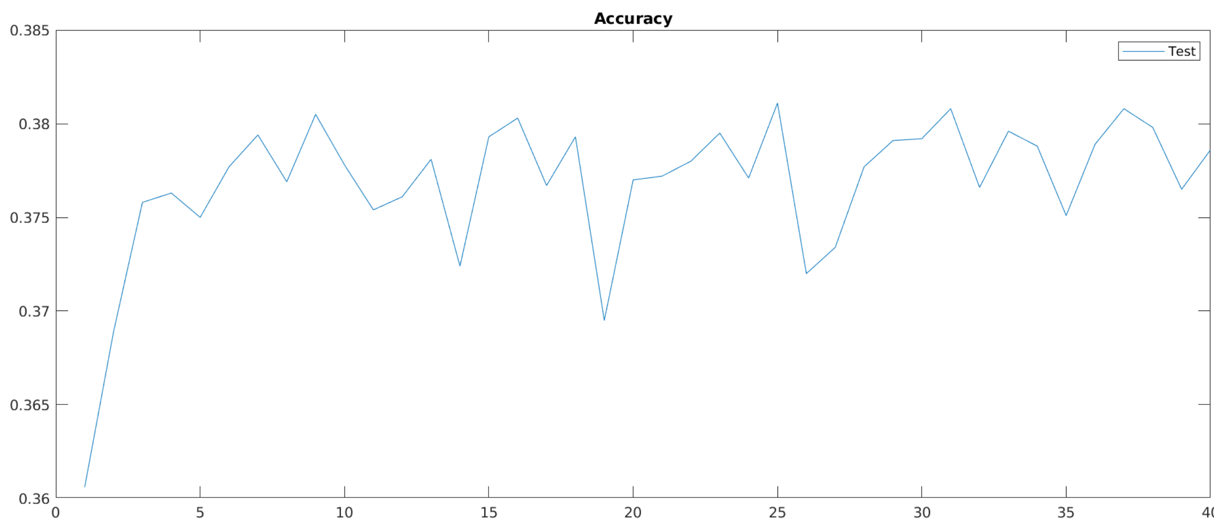


Figure 2: Improvement (i) Accuracy (lambda=1, n_epochs=40, n_batch=100, eta=0.001)

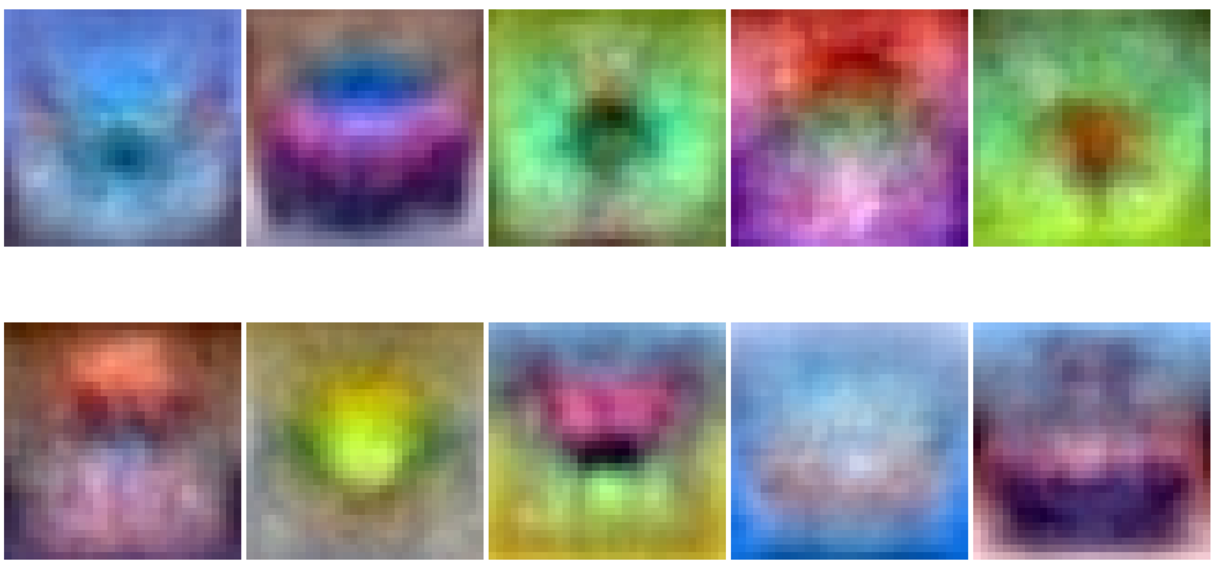


Figure 3: Improvement (i) Weights (lambda=1, n_epochs=40, n_batch=100, eta=0.001)

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

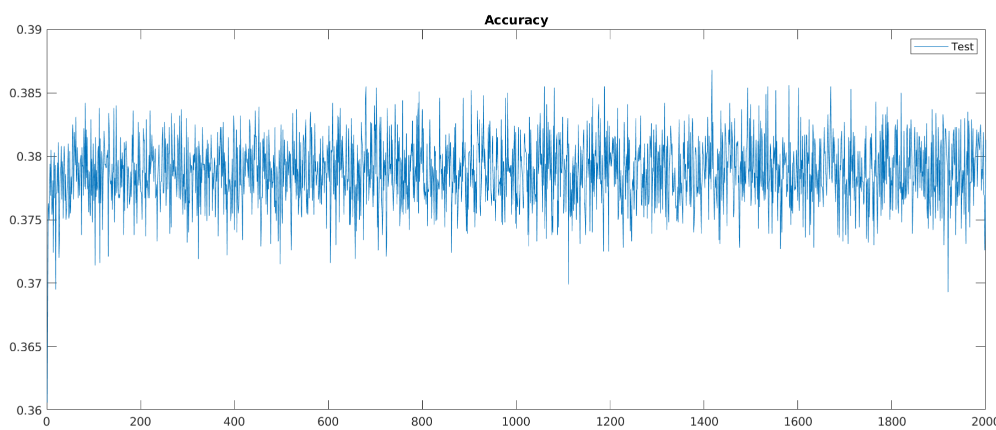


Figure 4: Improvement (ii) accuracy after 2000 epochs

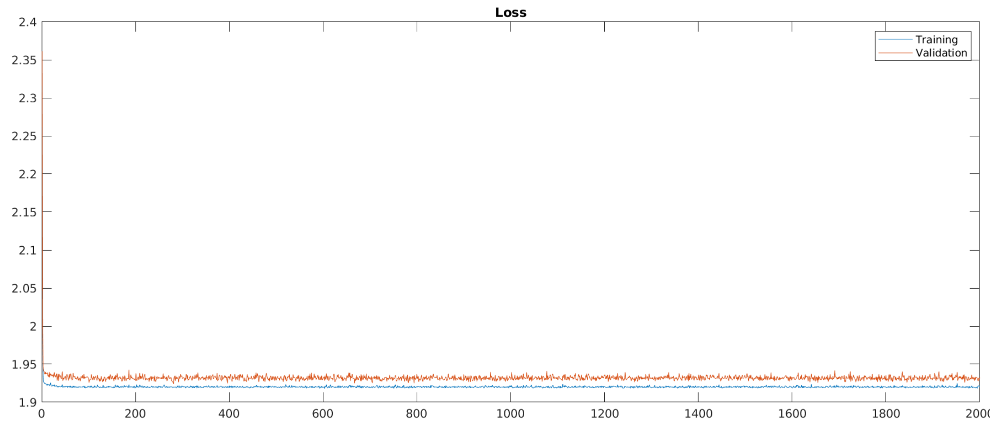


Figure 5: Improvement (ii) Loss after 2000 epochs

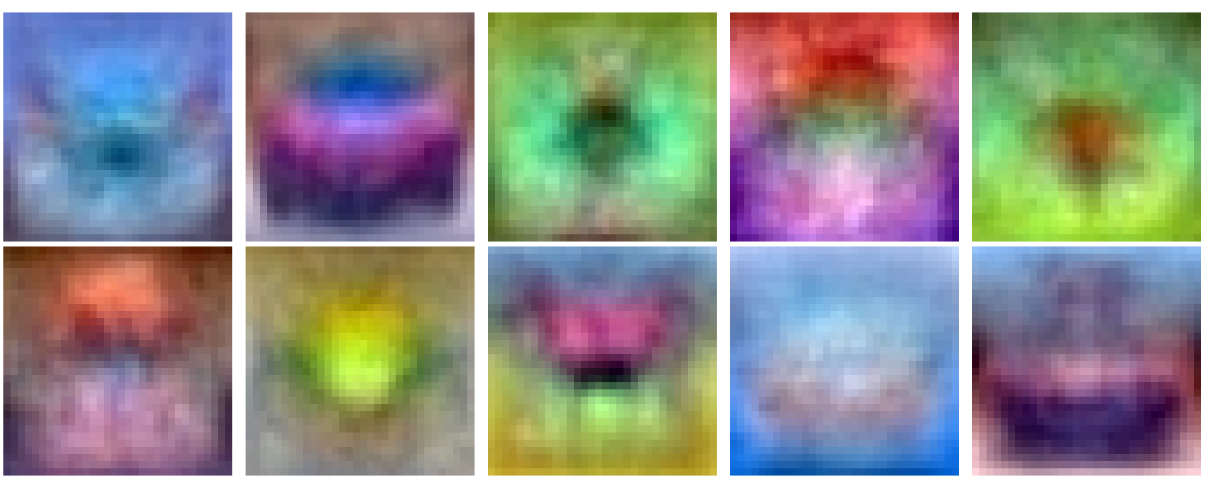


Figure 6: Improvement (ii) weights after 2000 epochs

1.3 Results Improvement (iii)

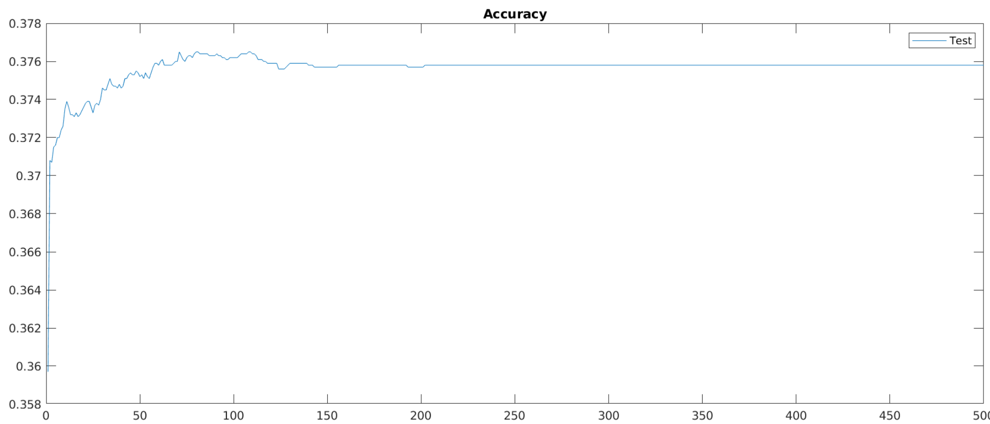
All the previous experiments were performed with shuffling the order of the training set. So in this experiment I will not shuffle the order to observe if it has a visible effect on the outcome. To switch on shuffling in the training loop I use $for j = randperm(n/gd\_params.n\_batch)$ and to switch it off I use $for j = 1 : n/gd\_params.n\_batch$.
As we can see in the figures below not shuffling the training data results in much smoother accuracy and loss curves.
The training results are not very different. Maybe we could conclude that shuffling the training data results in a slightly higher accuracy but the differences are very small.

epoch	**training loss**	**validation loss**	**accuracy**
1	2.35301	2.35979	35.97%
10	1.92442	1.93682	37.35%
50	1.92094	1.93256	37.52%
100	1.92058	1.93186	37.62%
200	1.92056	1.93169	37.57%
300	1.92056	1.93168	37.58%
400	1.92056	1.93168	37.58%
500	1.92056	1.93168	37.58%

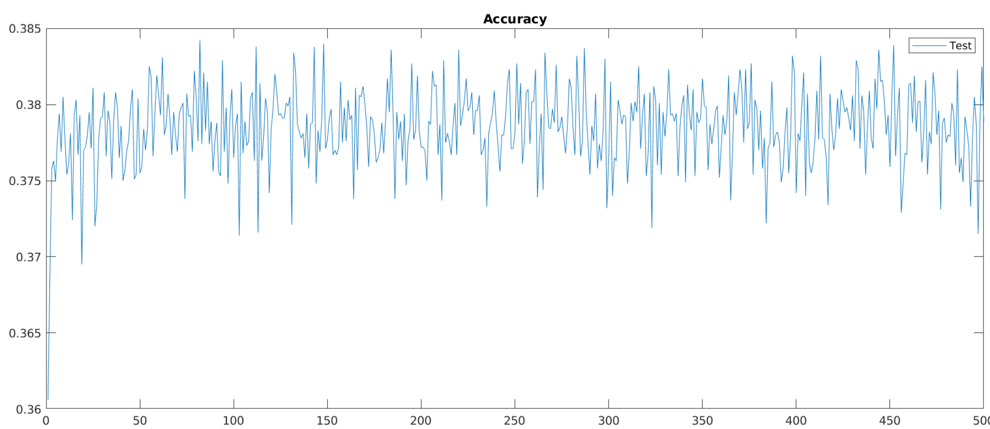
Table 2: Summary - not shuffled

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%

Table 3: Summary - shuffled

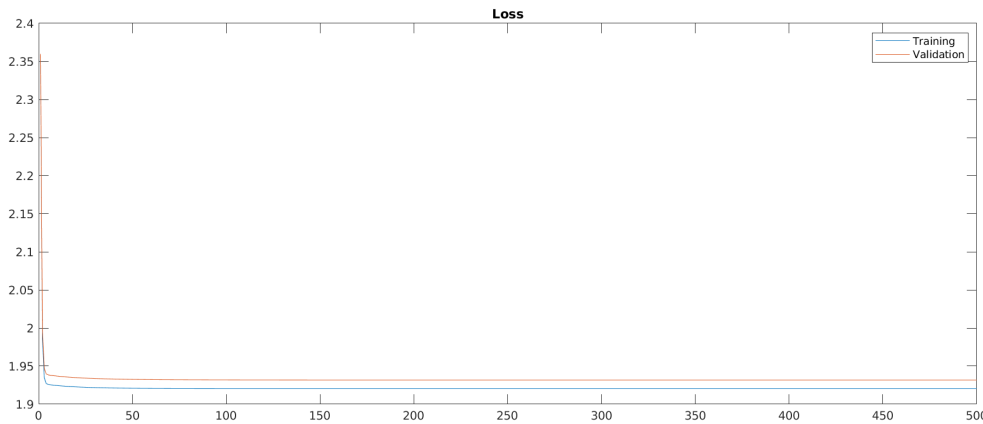


(a) shuffle disabled

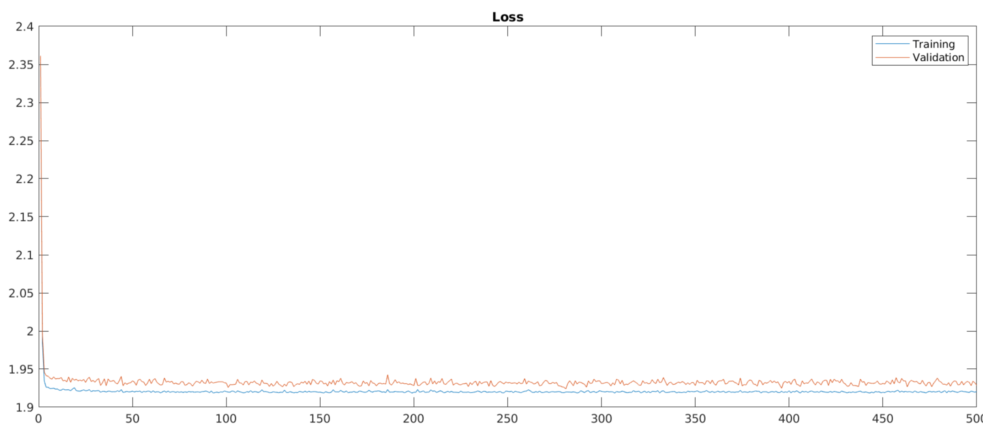


(b) shuffle enabled

Figure 7: Improvement (iii) Accuracy



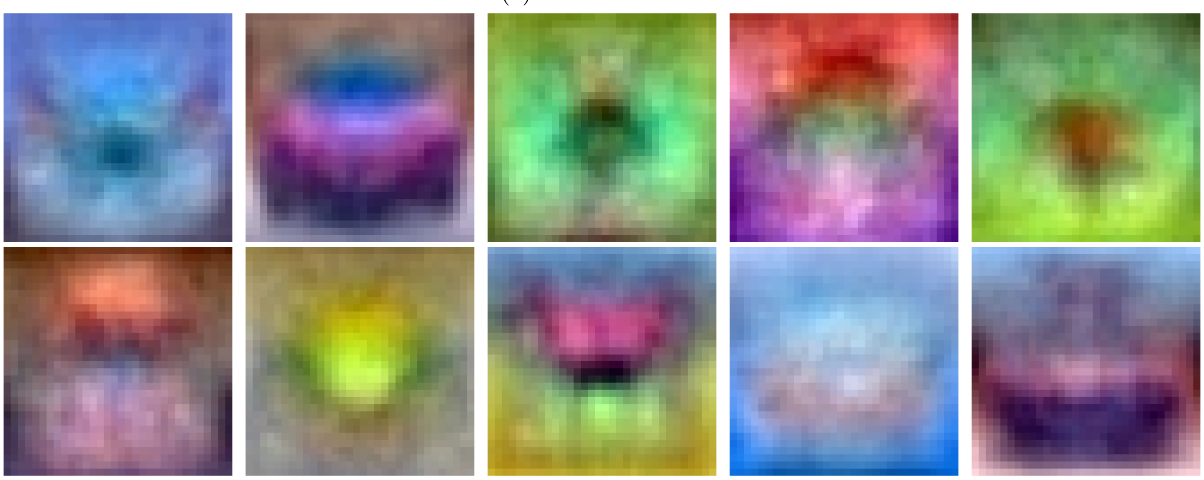
(a) shuffle disabled



(b) shuffle enabled

Figure 8: Improvement (iii) Loss

(a) shuffle disabled



(b) shuffle enabled

Figure 9: Improvement (iii) Weights