Hannes Kindbom

DD2424 Deep Learning – Assignment 2

Testing Gradient Implementation

To test and confirm that the analytical computations of the gradient were correct, the analytical gradient was compared to the numerical version as in assignment 1 (computed with the centered difference method). The relative error was computed, according to the formula in figure 1, for four different number of images and image dimensions. The results of the tests are shown in table 1.

$$\frac{|g_a - g_n|}{\max(\text{eps}, |g_a| + |g_n|)}$$

Figure 1. The relative error between a numerically computed gradient value g_n and an analytically computed gradient value g_a

Gradient test nr	Nr of images	Image dimension	λ	Relative error (W1, W2)	Relative error (b1, b2)
1	10	100	0	2.599e-11, 6.587e-11	3.6154e-11, 1.0214e-09
2	2	200	0	2.4913e-11, 5.1252e-11	1.3910e-11, 3.0614e-10
3	10	100	0.5	3.0692e-11, 1.4858e-11	7.0388e-11, 1.0233e-09
4	2	200	0.5	2.5921e-11, 1.9747e-11	2.5552e-11, 3.0680e-10

Table 1. The conducted gradient tests. Note that step length of h=0.0001 was used in the numerical computations.

One may observe that the relative error is < 1e-7 in all test cases, which indicate that the gradient implementation is correct.

To further check if the gradient implementation was correct, the network was trained on a small amount of training data (100 images) without regularization (λ = 0) for 300 epochs with batch size 100 and learning rate 0.001. The resulting training- and validation loss is shown in figure 2. As can be observed, the overfitting is significant, which indicate that the gradient computations and mini-batch gradient decent algorithm are both okay.

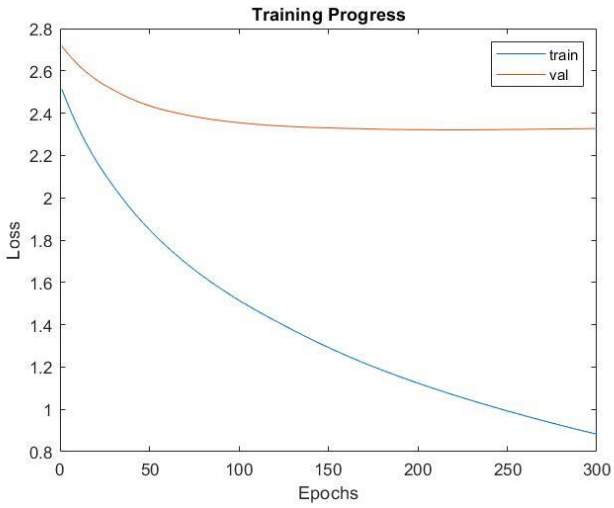

Figure 2. Attempting to overfit the training data as evidence for correct gradient implementation

Cyclical Learning Rate

To try to speed up training and avoid time-consuming searches for good values of the learning rate η, a cyclical learning rate was implemented. It was implemented according to equation (14) and (15) in the problem instructions. The cyclical learning rate was thereafter ran with two different parameter settings:

Setting 1: eta_min = 1e-5, eta_max= 1e-1, n_s=500, batch size 100, λ = 0.01, 1 cycle

Setting 2: eta_min = 1e-5, eta_max= 1e-1, n_s=800, batch size 100, λ = 0.01, 3 cycles

The resulting graphs (with plotting ten times per cycle) for setting 1 were:

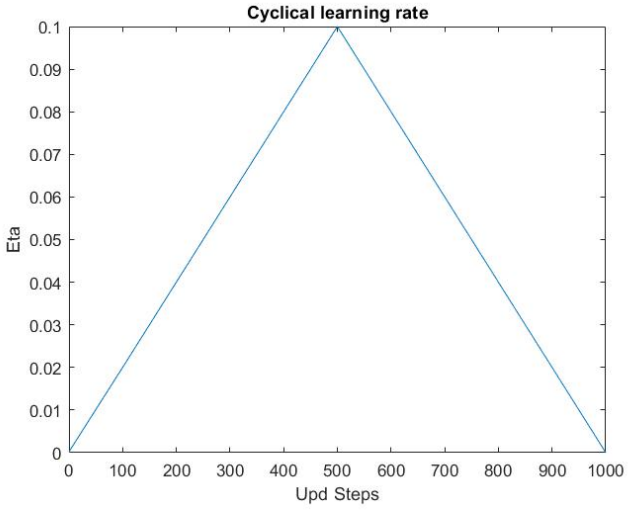

Figure 3. One cycle with step size n_s = 500 of the triangular learning rate

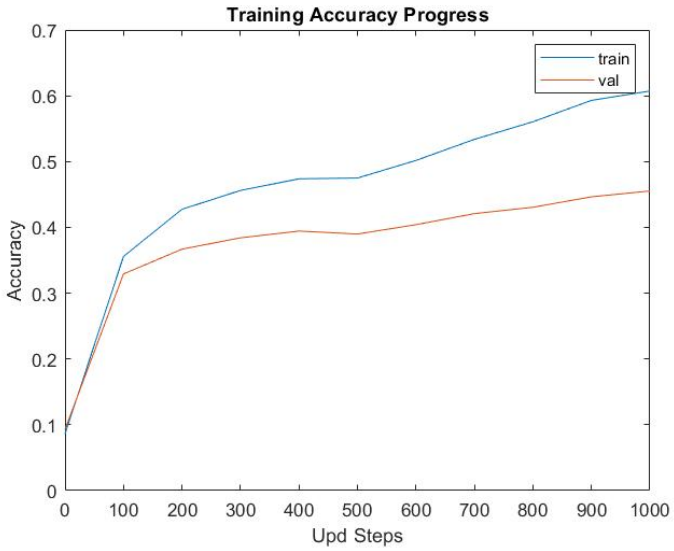

Figure 4. The training- and validation accuracy with hyperparameters according to setting 1. The achieved test accuracy was 45.88 %

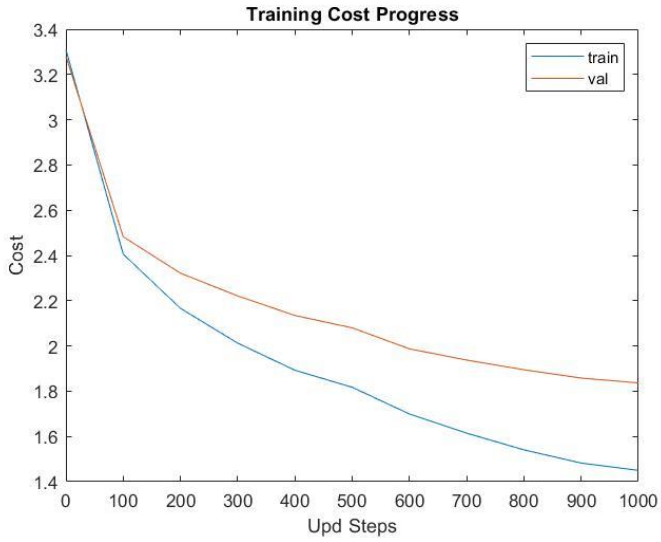

Figure 5. The progress of the cost during training with hyperparameters according to setting 1

The resulting graphs for setting 2 were:

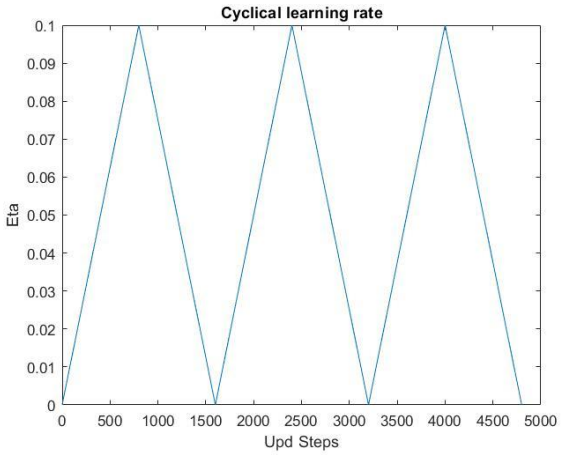

Figure 6. Three cycles with step size n_s = 800 of the triangular learning rate

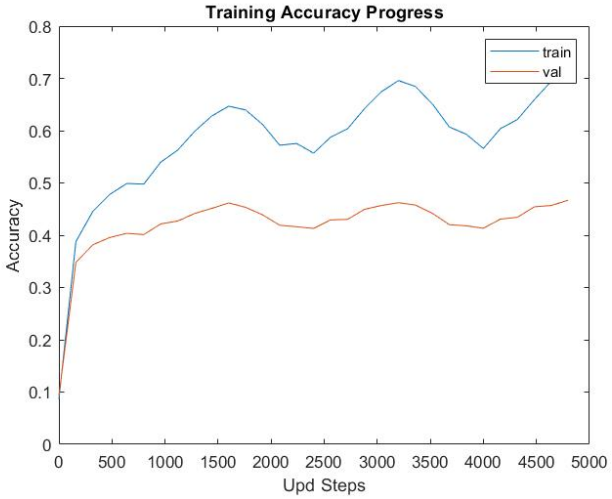

Figure 7. The training- and validation accuracy with hyperparameters according to setting 2. The achieved test accuracy was 46.77 %

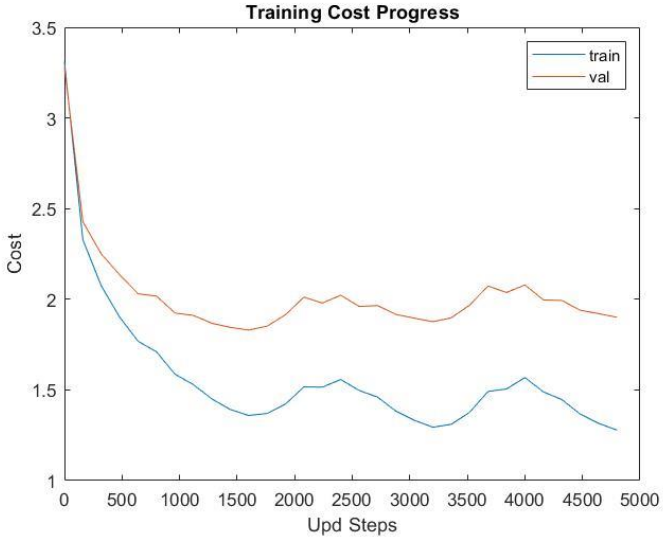

Figure 8. The progress of the cost during training with hyperparameters according to setting 2

The curves are similar to the ones in figure 3 and 4 in the problem instructions, as they should be. Furthermore, some interesting observations, based on these two experiments can be made. Smith writes in the original paper on learning rates, that increasing the learning rate periodically can achieve long term positive effects to the cost of short term negative effects.[1] In both experiments, this short term negative effect can be seen by the fact that the training- and validation accuracy decrease when the learning rate is maximum. The long term positive effect can be seen by the fact that, at least on the training data, the moving average accuracy increases. The long term benefits, may be explained by that occasionally larger learning rates enable the network to "jump out" of sharp local minima and converge to a new one. That is why the algorithm should be terminated when the learning rate is minimum.

Random Search

In order to optimize the performance of our network, a random search was then performed. The validation accuracy was registered while the value of λ varied. A searched was performed in three rounds, each with decreasing range of the λ values. All 5 training batches of the *cifar-10 dataset* were used for training, except for 5000 images in a validation set. One cycle of training with n_s = 2*floor(n / batch size), where n is the number of samples in the training set, was conducted for each λ value. The search for λ was done on a log scale according to the following randomized sampling algorithm:

```
l = l_min + (l_max - l_min)*rand(1, 1);
lambda = 10^l;
```

Round 1

Eight different λ values with l_min=-5 and l_max=-1, i.e. on the range [0.00001, 0.1] yielded:

Validation Accuracy	λ
0.5026	4.888099E-5
0.5046	0.002167
0.5026	0.000572
0.5046	0.002559
0.509	0.000822
0.5044	3.006148E-5
0.4542	0.036570
0.5032	0.001325

Table 2. Round 1 yielded maximum validation accuracy of 50.90 %. The best values found are highlighted in green.

Round 2

[1] Smith, L. N. (2015). Cyclical learning rates for training neural networks. arXiv:1506.01186 [cs.CV]

Based on the green highlighted values in round 1, a search on the range [0.0001, 0.01] for 12 different λ values, yielded the following result:

Validation Accuracy	λ
0.5014	0.000221
0.5022;	0.001472
0.501	0.000757
0.5014	0.001510
0.505	0.000907
0.5018	0.000173
0.5034	0.006047
0.5068	0.001151
0.5062	0.003482
0.5034	0.001002
0.5042	0.006969
0.5024	0.000174

Table 3. Round 2 yielded maximum validation accuracy of 50.68 %. The best values found are highlighted in green.

Round 3

Based on the green highlighted values in round 2, a search on the range [0.001, 0.01] for 10 different λ values, yielded the following result:

Validation Accuracy	λ
0.5074	0.001487
0.502	0.003837
0.5048	0.002751
0.5026	0.003910
0.505	0.003011
0.5014	0.001317
0.5016	0.007777
0.5038	0.003393
0.5054	0.005901
0.5042	0.003166

Table 4. Round 3 yielded maximum validation accuracy of 50.74 %. The best values found are highlighted in green.

The top three scores found were:

Validation Accuracy	λ
0.509	0.000822
0.5074	0.001487
0.5068	0.001151

Final Training and Evaluation

The highest achieved validation accuracy of 50.90 % was obtained for λ = 0.000822 in round 1. Therefore, the network was trained using that λ for 3 cycles using all the training data (50 0000 images) apart from 1000, which were used as a validation set. A full specification of the hyperparameters follows:

eta_min = 1e-5, eta_max= 1e-1, n_s= 2*floor(n / batch size)=980, batch size = 100, λ = 0.000822, 3 cycles, 50 hidden nodes

This setting yielded a test accuracy of 51.04 % and the following graphs:

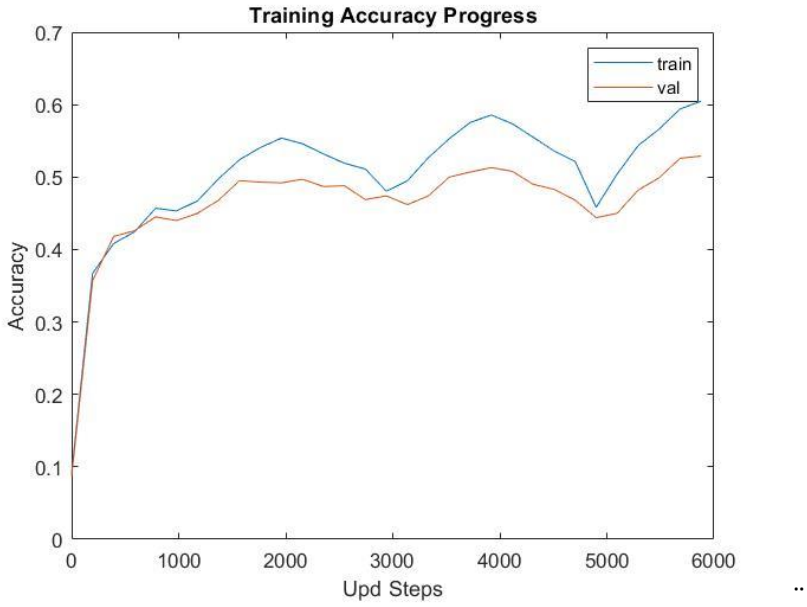

Figure 9. The training- and validation accuracy for the final model. The achieved test accuracy was 51.04 %

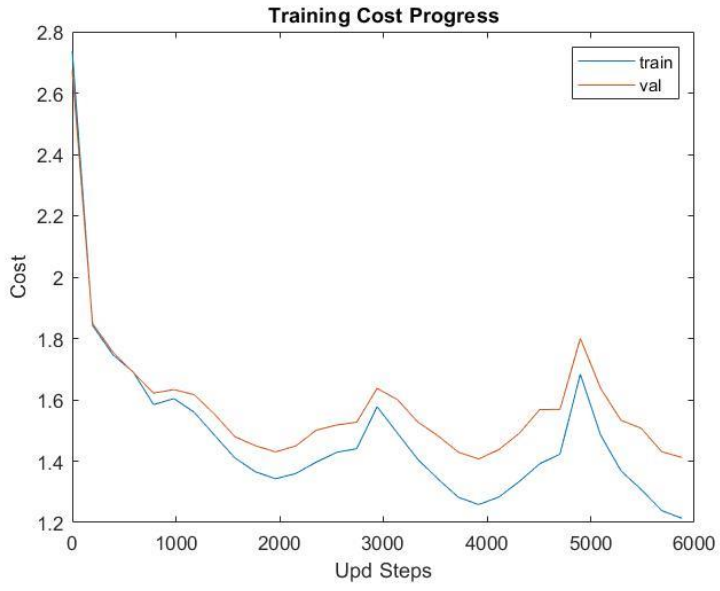

Figure 10. The training- and validation cost for the final model