DD2424
Deep Learning in Data Science
Assignment 2

Franco Ruggeri, fruggeri@kth.se

May 6, 2020

1 Gradients computation

I successfully managed to write the functions to correctly compute the gradients analytically. To check it, I compared the results with the numerical estimations of the gradients given by the function *ComputeGradsNumSlow* for the following test cases:

1. First mini-batch of size 1, no regularization ($\lambda = 0$) and first 20 dimensions.
2. First mini-batch of size 20, no regularization ($\lambda = 0$) and first 20 dimensions.
3. First mini-batch of size 20, $\lambda = 1$ and first 20 dimensions.

The results are shown in tables 1-4. As we can see, the differences are really small and this confirms the correctness of the analytical computations.

2 Cyclical learning rates

Figure 1 shows the learning curves and the learning rates for the default settings suggested in the instructions. The figures are very similar[1] to the ones provided

[1]Not exactly equal due to the different seed to the random number generator.

Test case	Max absolute difference	Max relative difference
1	4.14e-11	5.43e-07
2	7.02e-11	4.50e-07
3	1.85e-10	.85e-06

Table 1: Tests on gradients $\frac{\partial J}{\partial W_1}$.

Test case	Max absolute difference	Max relative difference
1	2.77e-11	1.33e-07
2	8.32e-11	5.38e-06
3	1.56e-10	5.36e-08

Table 2: Tests on gradients $\frac{\partial J}{\partial W_2}$.

Test case	Max absolute difference	Max relative difference
1	1.98e-11	1.53e-09
2	3.58e-11	7.93e-08
3	9.38e-11	4.23e-07

Table 3: Tests on gradients $\frac{\partial J}{\partial b_1}$.

Test case	Max absolute difference	Max relative difference
1	1.71e-11	8.01e-11
2	5.33e-11	1.49e-08
3	7.55e-11	2.41e-08

Table 4: Tests on gradients $\frac{\partial J}{\partial b_2}$.

in the instructions, confirming the absence of bugs in the implementation of *mini-batch gradient descent with cyclical learning rates.*

The effect of cycling learning rates can be seen especially in the second case (figures on the right). Ignoring the first cycle, which is affected by the initialization, during the first half of a cycle the parameters tend to go away from the local minimum (divergence, too high learning rates), while during the second half they go towards a local minimum (low learning rates). At the end of each cycle the network is in the vicinity of a local optimum, which is probably different from the previous ones.

3 Coarse-to-fine random search

To optimize the network, I performed a coarse-to-fine random search to find a good value for the amount of regularization, using cross-validation with a validation set of 5000 images. All the available data was used. In particular:

- Stage 1: coarse random search with 10 values uniformly sampled on a log scale between 10^{-5} and 10^{-1}. I run training with $n_{cycles} = 2$ and $n_s = 2\ floor(n/n_{batch})$. The best 3 values are reported in table 5.
- Stage 2: finer random search with 10 values uniformly sampled on a log scale between 10^{-4} and 10^{-2} (narrower range including the 3 best values from stage 1). I run training with $n_{cycles} = 3$ and $n_s = 2\ floor(n/n_{batch})$. The best 3 values are reported in table 6.

4 Final network

After the random search, I trained a network with $\lambda = 0.002584$ (best value found), $n_{cycles} = 3$, $n_s = 2\ floor(n/n_{batch})$ and a validation set of 1000 images. I got an accuracy of 52.30%. The learning curves are shown in figure 2.

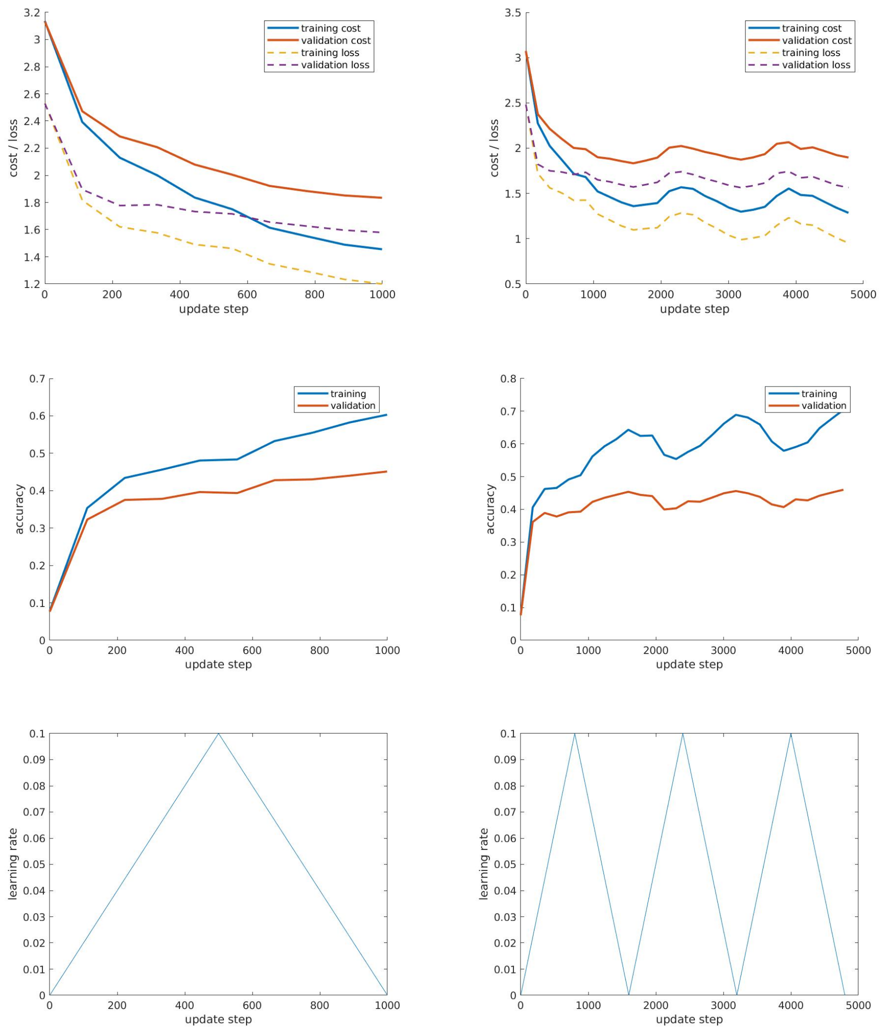

Figure 1: Learning curves and cycling learning rates for $n_s = 500$, $n_{cycles} = 1$ (on the left) and $n_s = 800$, $n_{cycles} = 3$ (on the right).

λ	Accuracy (%)
0.006679	52.08
0.002193	52.34
0.000212	52.78

Table 5: Best 3 values for λ found with coarse random search. Settings: $l_{min} = -5$, $l_{max} = -1$, $n_{trials} = 10$, $n_{cycles} = 2$.

λ	Accuracy (%)
0.006525	52.32
0.007151	52.64
0.002584	52.70

Table 6: Best 3 values for λ found with fine random search. Settings: $l_{min} = -4$, $l_{max} = -2$, $n_{trials} = 10$, $n_{cycles} = 3$.

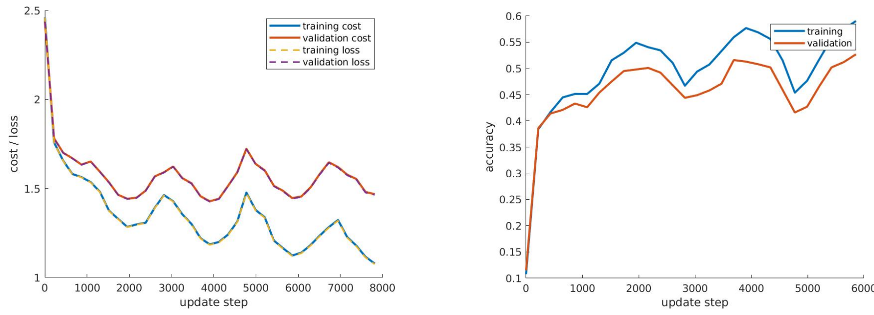

Figure 2: Learning curves for the final network. Settings: $\lambda = 0.002584$, $n_{cycles} = 3$, $n_s = 2\ floor(n/n_{batch})$.