DD2424 Deep Learning in Data Science

Assignment 2

Clara Tump *tump@kth.se*

April 26, 2019

1 Introduction

This assignment consisted of implementing a one-layer Neural Network.

2 Methods

A 2-layer neural network was implemented from scratch using Python 3.6. The neural network is trained using backpropagation and minibatch gradient descent. As a data set CIFAR-10 is used, which consists of 32x32 images corresponding to 10 classes of objects. The model makes use of cyclical learning rate, and the value for the regularization parameter λ is optimized using course and fine search.

3 Results

3.1 Gradients

The correctness of the analytically calculated gradients was checked using two numerical gradient calculation methods for comparison. Firstly, the gradients were checked against the finite difference method, these results are visible in Figure **??** . In addition, the analytical gradients are compared with the gradients computed using the centered difference method. This method is somewhat slower than the finite difference method, but more accurate. These results are give in Figure **??**. In both comparisons it is visible that the analytically computed gradients are very similar to the numerical ones, which gives strong evidence that the gradient computation is correct.

3.2 Cyclic learning rate

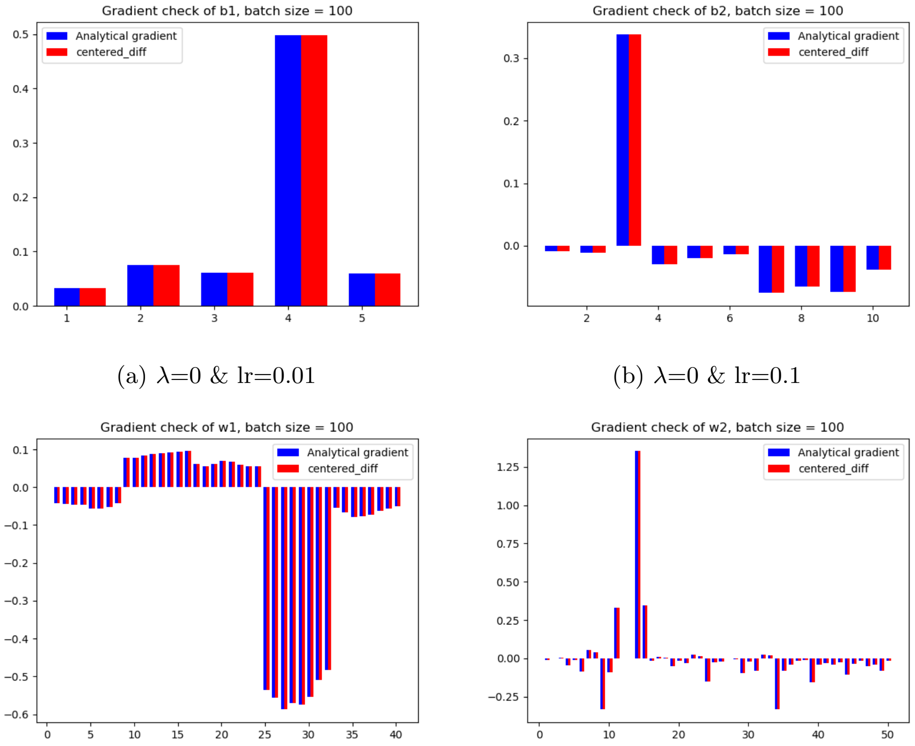
A cyclic learning rate was implemented. This means that the learning rate increases and decreases linearly between η_{min} and η_{max} over a number of update steps (batch updates within an epoch) (see Figure 3). In this case, η_{min} was set to $1e-5$ and η_{max} was set to $1e-1$. One cycle of the learning rate is the number of epochs it takes for the learning rate to increase from η_{min} to η_{max} and back to η_{min} again.

Figure 4 shows the loss and the accuracy for 1 cycle and 3 cycles of the learning rate. They show that the loss decreases at the beginning of a cycle (when the learning rate increases). In the case of the 3 cycles the loss even increases slightly at the end of a cycle (when the learning rate decreases). The opposite effect is visible for the accuracy. This indicates that most learning happens at the beginning of the cycle, and sometimes there is even a negative effect on the learning at the end of the cycle.

3.3 Optimization of parameters

Course and fine search were used for the optimization of hyperparameters λ. In the course search, a broad range of values were tested, and their performance on the validation set were recorded. 14 epochs were used to train the models. 10 values were tested: $1e^k$ with $k \in \{-9, -8, -7, -6, -5, -4, -3, -2, -1, 0\}$. The results are given in Table 1

As can be seen in Table 1, the performance is the best when $\lambda = 1^{-7}$ and when $\lambda = 1^{-}3$. Subsequently, the fine search explored the performance on values of λ around these high performance values in the course search. The results of the fine search are given in Table 2. These results show the best performance with $\lambda = 6e-4$, $\lambda = 8e-4$, and $\lambda = 1e-3$. Thinking of robustness, the middle value of these is chosen for the final model. This model is run for 32 epochs with $\lambda = 8e-4$. The final accuracy of this model on the test set is **0.5212**. The loss and accuracy of this model is plotted in Figure **??**



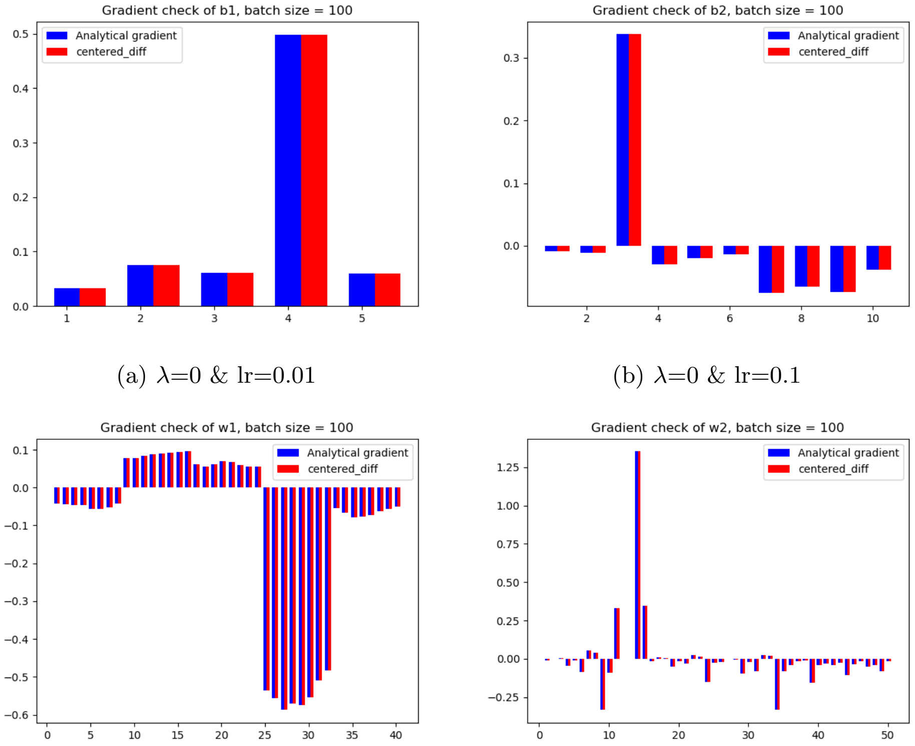
(a) λ=0 & lr=0.01

(b) λ=0 & lr=0.1

(c) λ=0.1 & lr=0.01

(d) λ=1 & lr=0.01

Figure 1: Gradient check with the centered difference method



(a) λ=0 & lr=0.01

(b) λ=0 & lr=0.1

(c) λ=0.1 & lr=0.01

(d) λ=1 & lr=0.01

Figure 2: Gradient check with the finite difference method

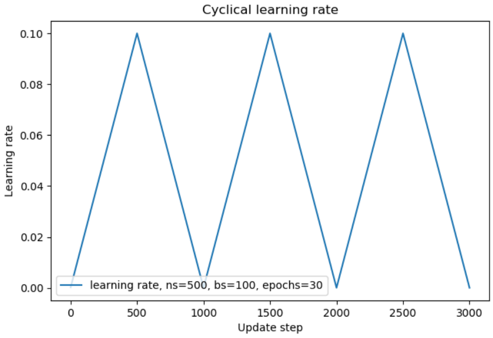
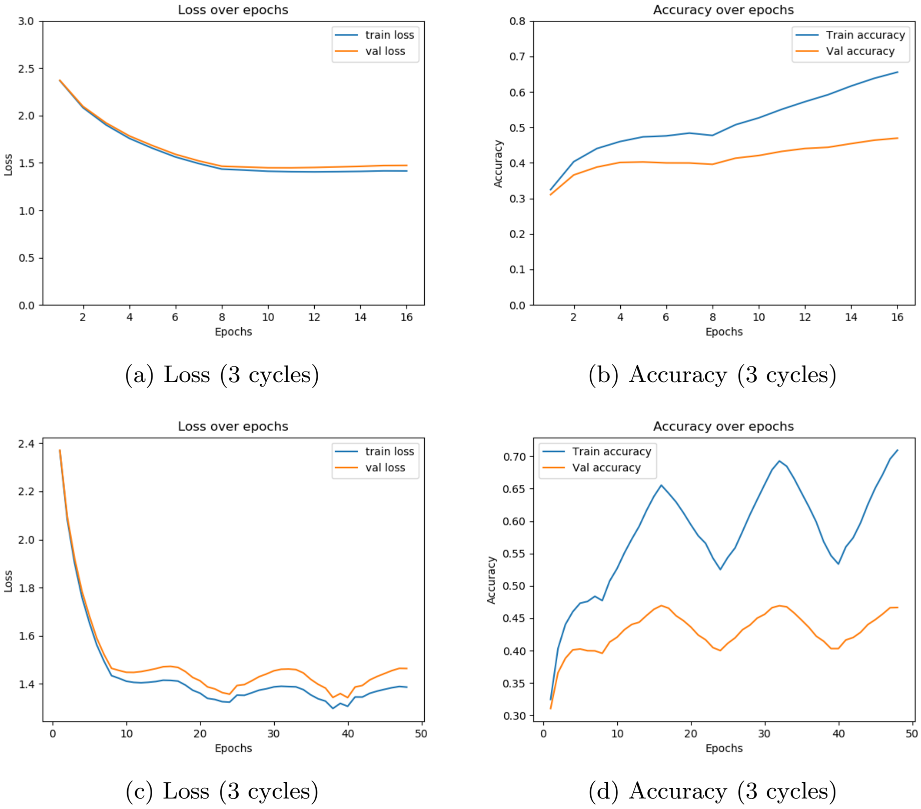


Figure 3: Cyclic learning rate



(a) Loss (3 cycles)

(b) Accuracy (3 cycles)

(c) Loss (3 cycles)

(d) Accuracy (3 cycles)

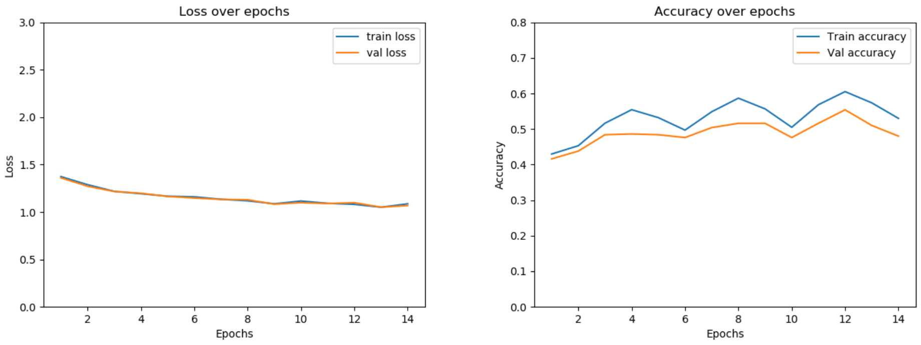
Figure 4: Loss and accuracy using cyclic learning rate

k	val acc
-9	0.492
-8	0.502
-7	0.52
-6	0.514
-5	0.488
-4	0.498
-3	0.526
-2	0.496
-1	0.37
0	0.19

Table 1: Results course search

λ	val acc
$2e-8$	0.504
$4e-8$	0.526
$6e-8$	0.508
$8e-8$	0.514
$1e-7$	0.506
$2e-7$	0.474
$4e-7$	0.484
$6e-7$	0.506
$8e-7$	0.498
$2e-4$	0.524
$4e-4$	0.512
$6e-4$	0.536
$8e-4$	0.534
$1e-3$	0.534
$2e-3$	0.496
$4e-3$	0.524
$6e-3$	0.524
$8e-3$	0.506
0	0.19

Table 2: Results fine search



(a) Loss

(b) Accuracy

Figure 5: Final model with optimized λ