Deep Learning in Data Science, DD2424

Short report on assignment 2

Image classification with a two-layer network

Côme Lassarat

April 16, 2022

1 Introduction

This work aims to implement a two-layer network to classify images from the CIFAR-10 dataset. The network designed will be trained using mini-batch gradient descent and cyclical learning rates. First, the tests ran to verify if the different functions (such as the gradients computation) are correct will be presented. In a second place, the different scenarios with relevant plots will be described.

2 Implementation checking

One of the main difficulty of this assignment is to correctly implement the analytical gradients calculation. In order to do so, a separate function were used to compute the numerical gradients. Then, the relative error (element wise) between the numerical gradients and the analytical gradients (implemented in the network) was calculated on a batch from the dataset used. Finally, the maximum of these relative errors is analyzed, for different lambda settings (and $\eta = 0.001$) (Table 1). For every experience, the starting matrices W and b are always the same.

	W_1	b_1	W_2	b_2
Max relative gradient error	$2.92e-8$	$5.52e-9$	$7.29e-8$	$4.92e-10$

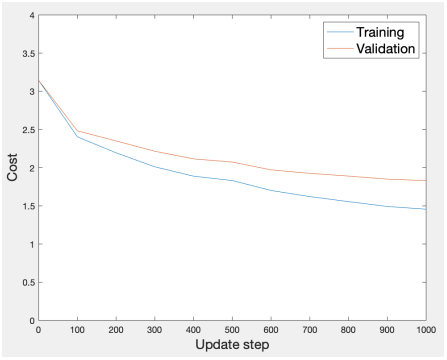
Tabell 1: Gradient checking - Maximum relative error for each gradient matrix ($\lambda = 0$)

These errors are smaller than $1e-7$: we can then assumed our gradient implementation is correct.

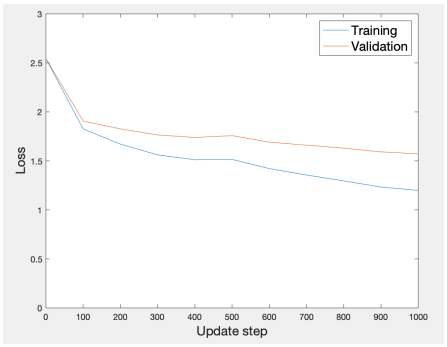
3 Training the neural network with cyclical learning rates

3.1 Training for one cycle

The training curves for one cycle training can be seen on Figure 1 and Figure 2, for the following configuration: $\eta_{min} = 10^{-5}$, $\eta_{max} = 10^{-1}$, $\lambda = 0.01$, $n_{batch} = 100$ and $n_s = 500$. The final test accuracy at the end of training is 46.75% (Table 2).

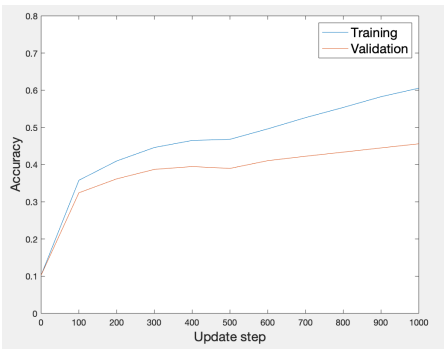


(a) Plot of the cost function over the update steps

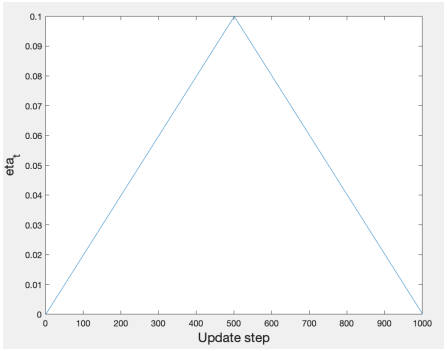


(b) Plot of the loss function over the update steps

Figur 1



(a) Plot of the accuracies over the update steps



(b) Plot of the loss function over the update steps

Figur 2

Test accuracy
46.00%

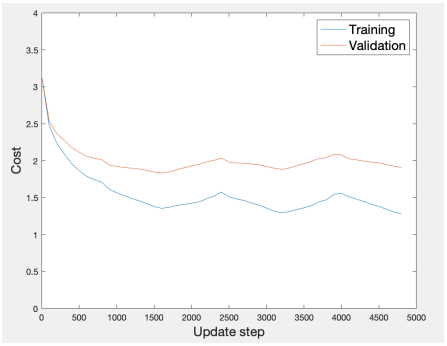
Tabell 2: Final accuracy

We can observe on the Figure 1 and Figure 2 that the gradient descent is effective (the cost and loss functions decrease and the accuracies increase). Furthermore, we can see that:

- at the beginning of the learning process, the cost and accuracy respectively decreases and increases quickly: this is because the learning rate η_t increase as the steps goes. Then, until a certain point, the gradient descent algorithm converges faster.
- when the cost and accuracies come close to being constant (around the 500th step), they then start to respectively to decrease and increase again: this matches to the fact that η_t reached its maximum and starts to decrease. As the learning rate is getting lower, this allow the network to jeep learning.

3.2 Training for three cycles

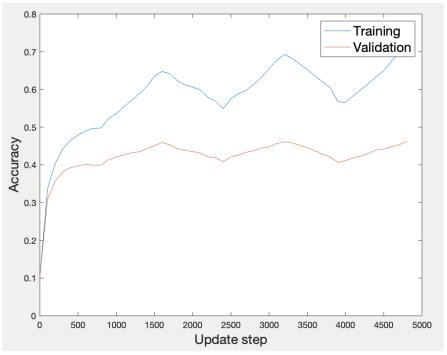
The training curves for one cycle training can be seen on Figure 3 and Figure 4, for the following configuration: $\eta_{min} = 10^{-5}$, $\eta_{max} = 10^{-1}$, $\lambda = 0.01$, $n_{batch} = 100$ and $n_s = 800$. The final test accuracy at the end of training is 46.75% (Table 3).



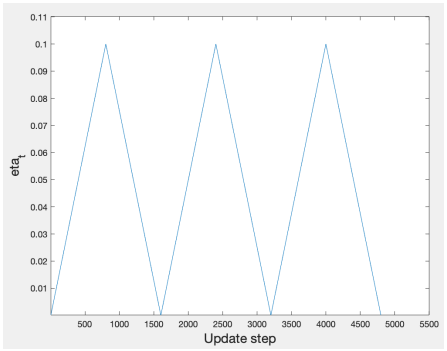
(a) Plot of the cost function over the update steps

(b) Plot of the loss function over the update steps

Figur 3



(a) Plot of the accuracies over the update steps



(b) Plot of the η over the update steps

Figur 4

Test accuracy
46.75%

Tabell 3: Final accuracy

The same observation as Section 3.2 can be made, only this time the phenomenons described are happening three times (because there is three cycles). As a result, the cost and accuracy respectively decreases and increases on average (Figure 3 and 3).

3.3 Coarse-to-fine random search to find λ

Given that our previous network training is working, the regularization term λ needs to be optimized

3.3.1 Coarse search

In order to do this, several values of λ will first be tried on a uniform log-scale from 10^{-5} to 10^{-5}. The network configuration is the following: $\eta_{min} = 10^{-5}$, $\eta_{max} = 10^{-1}$, $n_{batch} = 100$ and $n_s = 2floor\frac{n_{train}}{n_{batch}}$, $n_{cycles} = 2$

- Best λ values : $[10^{-5},\ 3.73e-5,\ 1.9e-3]$
- Best associated validation accuracy : $[52.10\%,\ 52.52\%,\ 52.66\%]$

3.3.2 Fine search

Finally, the lambda search is focused on a narrower range around the values of λ previously found, for 2 cycles.

- Best λ values: $[3.49e-3,\ 2.7e-3,\ 2.6e-3]$
- Best associated validation accuracy: $[52.36\%,\ 52.46\%,\ 52.96\%]$

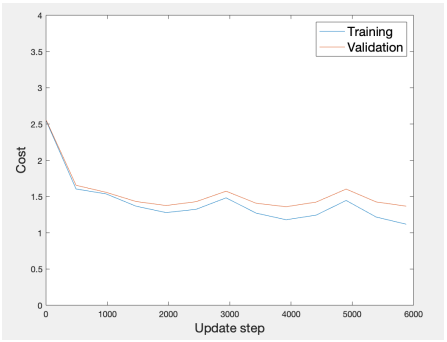
Thus, we found $\lambda_{best} = 2.6e-3$.

3.3.3 Training the final network

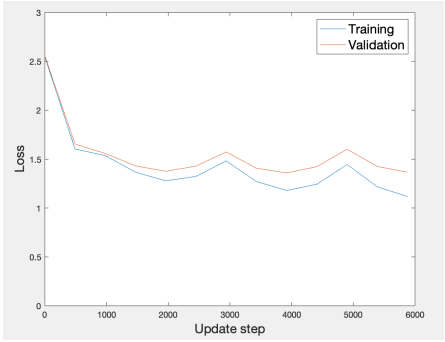
Finally, the network is trained using 3 cycles, $\lambda = \lambda_{best}$ and the same parameters as section 3.3. The plots are on 5 and 6. The final test accuracy is 50.75% (4).

Test accuracy
51.30%

Tabell 4: Final accuracy

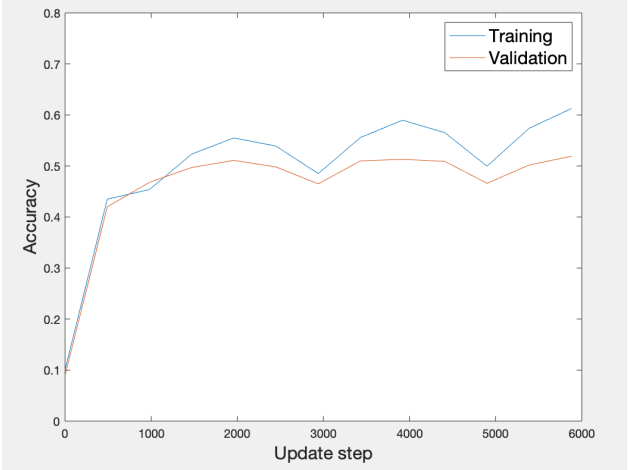


(a) Plot of the cost function over the update steps



(b) Plot of the loss function over the update steps

Figur 5



Figur 6: Plot of the accuracies over the update steps