# Deep learning homework 1

**Jovan Prodanov (63180376)**

## 1 Introduction

As stated in the homework, I implemented all the missing functionalities of the network and started testing it on the CIFAR-10 dataset. The answers to the stated subtasks of the 6th assignment called network training are answered in the following sections.

I just want to point out that I will present the classification accuracy as CA and it will be given in decimal points and not in percentage. Also for L2 regularization I will just say L2 from now on. Another thing would be LR the learning rate and DR the decay rate. In the 1 table, only the hidden layers are shown without the output and input layers (the input layer consisted of 3072 neurons and the output consisted of 10 neurons for the CIFAR-10 dataset I worked on).

For L2 I used $\lambda = 0.01$. For the adam optimization $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$

## 2 Tasks

### 2.1 Subtask 1

I used lower batch sizes and higher epoch times to get more accurate results, and the network also performed better with these parameters. I also experimented with only one hidden layer and higher batch size. Then, I also tried with more layers with different neuron numbers.

The best results I got so far was with a network with 2 hidden layers [200, 100] optimized with Adam and with L2 with a LR of 0.0002 and DR 0.01. It was trained for 30 epochs and with a mini-batch size of 16 and got 0.5164 CA. I played a lot with LR and found that 0.0002 is the "sweet spot", if I went below that it would become inaccurate. I also had some networks with smaller DR, but the default DR of 0.01 worked best for all my networks.

I also played with L2 and had the better results with it. During training, the losses were not very small, but the validation losses looked good.

### 2.2 Subtask 2

I left this task for last and got my best result so far. 3 table, the network optimized with Adam achieved the better result with 0.5164 CA, but at the cost of taking more time to train.

Adam was by far slower to train as it has to do more calculations to update weights and biases, but all in all it was more accurate.

### 2.3 Subtask 3

In one case, the Adam optimizer with L2 regularization was slightly better at 0.5087 CA, but had higher losses during training (but the validation losses seemed fine). On the other hand, without L2 it reached 0.5031 CA and everything looked normal. When the network was optimized at SGD, it achieved a slightly better result with L2 enabled. We can see these results in the table 1.

Therefore, we can assume that L2 regularization helps the network perform better.

### 2.4 Subtask 4

For this task, the results can be found in 2 and it seems that the exponential learning rate is associated with a better CA. The validation loss and loss throughout the training was also a little better (less), but the results were quite similar.

## 3 Tables

| *Num.* | *Hidden layers* | *Schedule* | *L2* | *Optimizer* | *LR* | *DR* | *Epochs* | *Batch size* | *CA* |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 200 | Exp. | No | SGD | 0.001 | 0.01 | 20 | 32 | 0.4887 |
| 2 | 200 | Exp. | Yes | SGD | 0.001 | 0.01 | 20 | 32 | 0.4812 |
| 3 | 200, 100 | Exp. | No | SGD | 0.001 | 0.01 | 20 | 16 | 0.4983 |
| 4 | 200, 100 | Exp. | Yes | SGD | 0.001 | 0.01 | 20 | 16 | 0.5007 |
| 5 | 200, 100 | Exp. | Yes | SGD | 0.001 | 0.01 | 30 | 32 | 0.4988 |
| 6 | 200, 100 | Exp. | Yes | Adam | 2e-5 | 0.001 | 30 | 16 | 0.4479 |
| 7 | 200, 100 | Exp. | Yes | Adam | 2e-4 | 0.001 | 30 | 16 | 0.5087 |
| 8 | 200, 120, 80, 50 | Exp. | Yes | Adam | 2e-4 | 0.01 | 30 | 16 | 0.4788 |
| 9 | 200, 100 | Exp. | No | Adam | 2e-4 | 0.001 | 30 | 16 | 0.5031 |
| 10 | 200,100 | Exp. | Yes | Adam | 2e-4 | 0.01 | 30 | 16 | 0.5164 |

Table 1: Results

| *Num.* | *Hidden l.* | *Schedule* | *L2* | *Optimizer* | *LR* | *DR* | *Epochs* | *Batch size* | *CA* |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 200, 100 | None | Yes | SGD | 0.001 | 0.01 | 30 | 16 | 0.4982 |
| 2 | 200, 100 | Exp. | Yes | SGD | 0.001 | 0.01 | 30 | 16 | 0.4991 |

Table 2: Rate scheduling table

| *Num.* | *Hidden l.* | *Schedule* | *L2* | *Optimizer* | *LR* | *DR* | *Epochs* | *Batch size* | *CA* |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 200, 100 | Exp. | Yes | SGD | 0.001 | 0.01 | 30 | 16 | 0.4979 |
| 2 | 200, 100 | Exp. | Yes | Adam | 0.0002 | 0.01 | 30 | 16 | 0.5164 |

Table 3: SGD vs Adam

## 4 Conclusion

To conclude, for this assignment I created and later on tested and evaluated a neural network. Used different approaches

to optimize and create better results from it. My computer specifications were pretty good so I had no problem running training on multiple networks as to speed up developing, but I was it got me thinking about the speed of these calculations on the GPU. So, I also learning why networks are trained on the GPU rather than the CPU.