# DD2424 Assignment 1

**Levent Güner**
**leventg@kth.se**

## Introduction

This assignment is about implementing one-layer Neural Network with mini batch gradient descent.

## Data

CIFAR-10 dataset is used, which has 10 class labels representing different objects and 10,000 different 32x32 images per class.

## Methods

There is a single layer neural network generated and tested with different configurations.

## Results

```
grad(W)
-------
Analytic grad(W):  0.08305772090258266
Numeric grad(W):  0.08305772090877461
Sum of absolute diff.:  1.6833892724221755e-08

grad(B)
-------
Analytic grad(b):  0.17941312011993713
Numeric grad(b):  0.17941312013824984
Sum of absolute diff.:  1.0613632472900036e-09
```

**Table 1: Gradient results**

I have checked the gradients with Centered Difference Formula and got pretty accurate results can be seen in Table 1.

| λ | lr(eta) | Training accuracy | Final testing accuracy |
|---|---|---|---|
| 0 | 0.1 | 0.28 | 0.24 |
| 0 | 0.01 | 0.42 | 0.37 |
| 0.1 | 0.01 | 0.34 | 0.33 |
| 1 | 0.01 | 0.23 | 0.22 |

**Table 2: Accuracies with different configurations**

Table 2 shows different $\lambda$ and learning rate configurations and their results. The best result is when $\lambda$=0 and learning rate is 0.01.

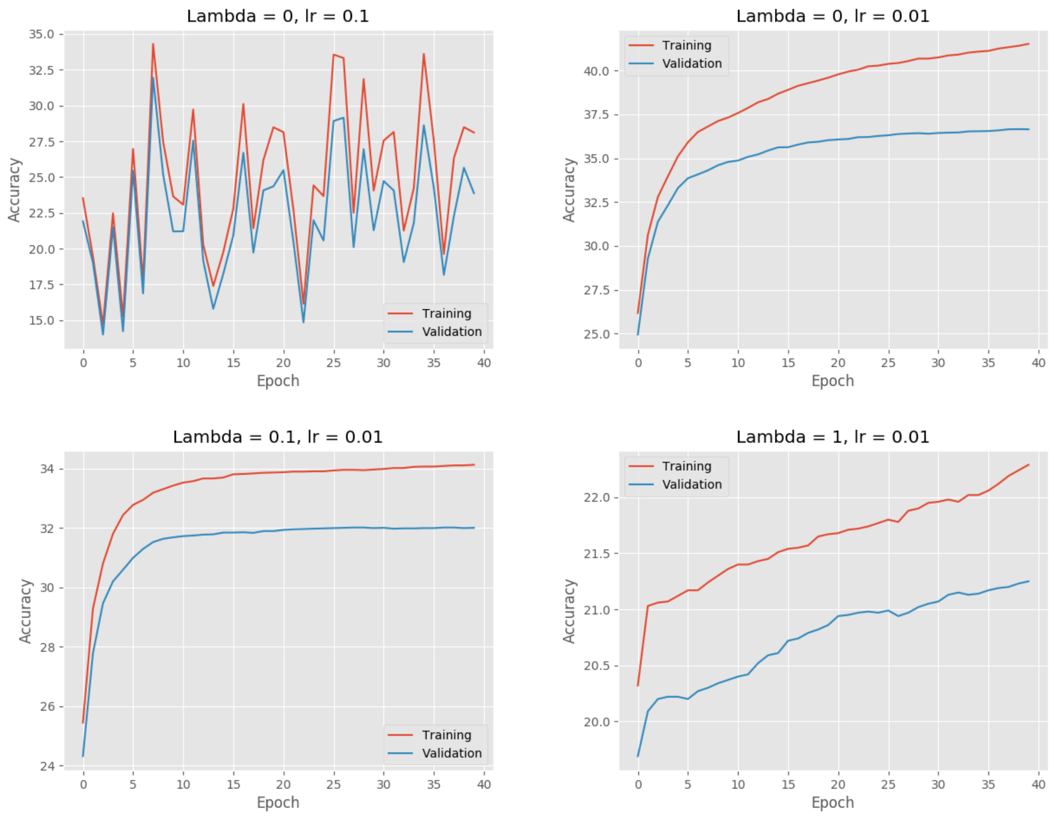



**Figure 1: Accuracies over epoch**

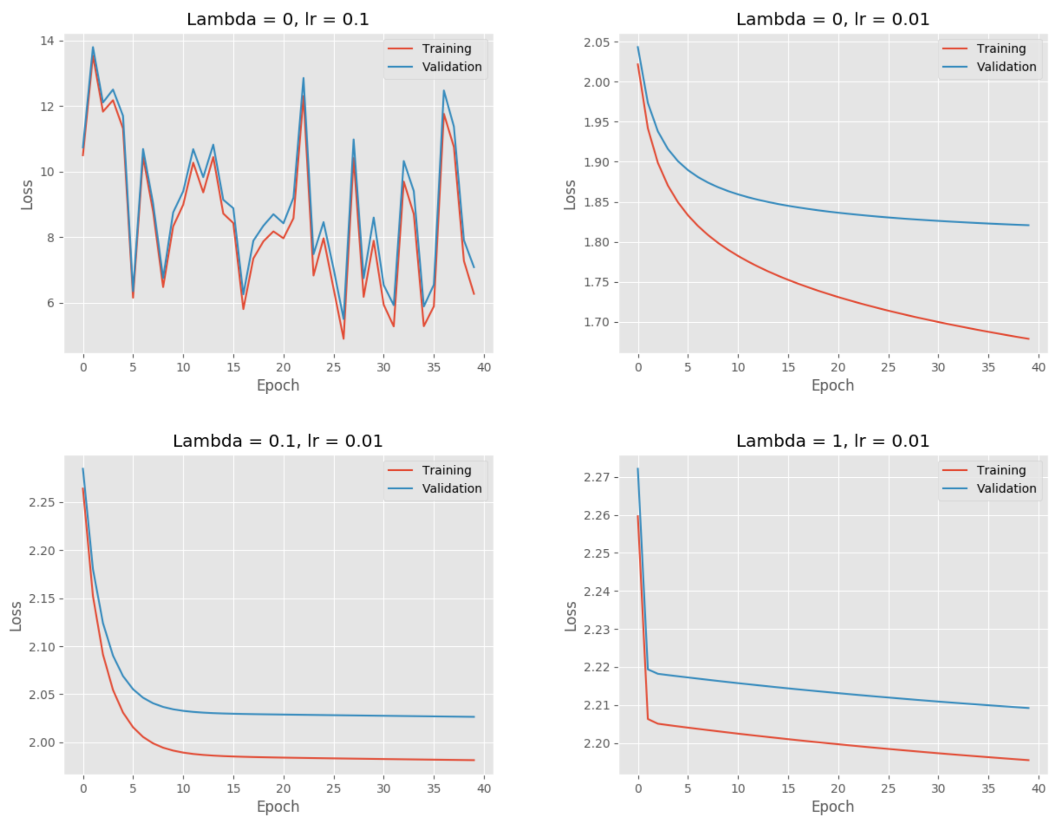
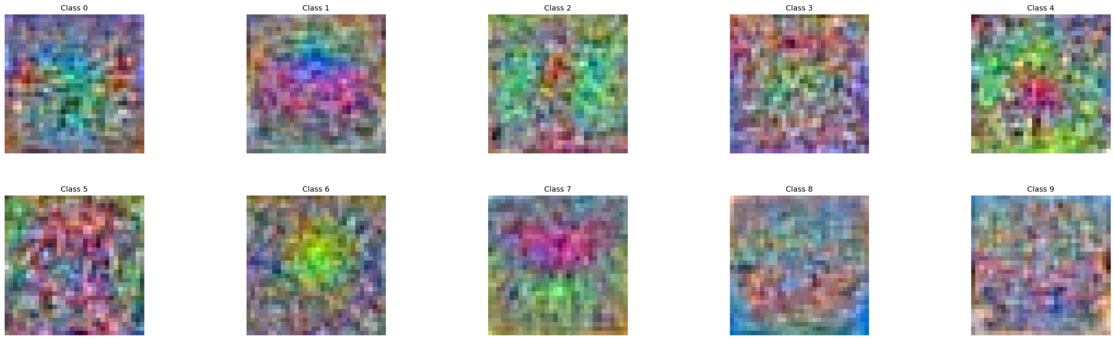


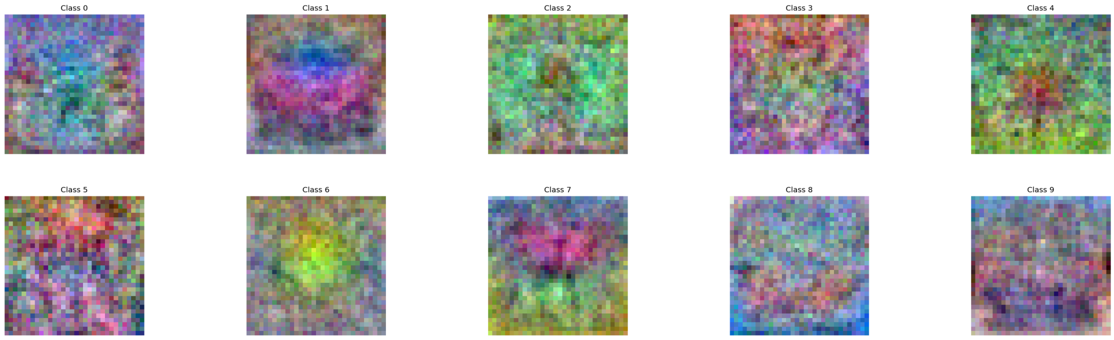

**Figure 2: Losses over epoch**

Levent Güner
leventg@kth.se

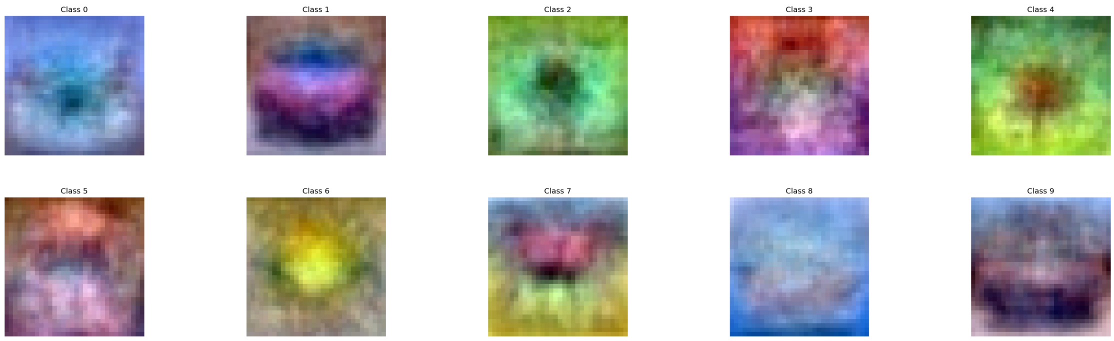
Lambda = 0, lr = 0.1




Lambda = 0, lr = 0.01




Lambda = 0.1, lr = 0.01




Lambda = 1, lr = 0.01

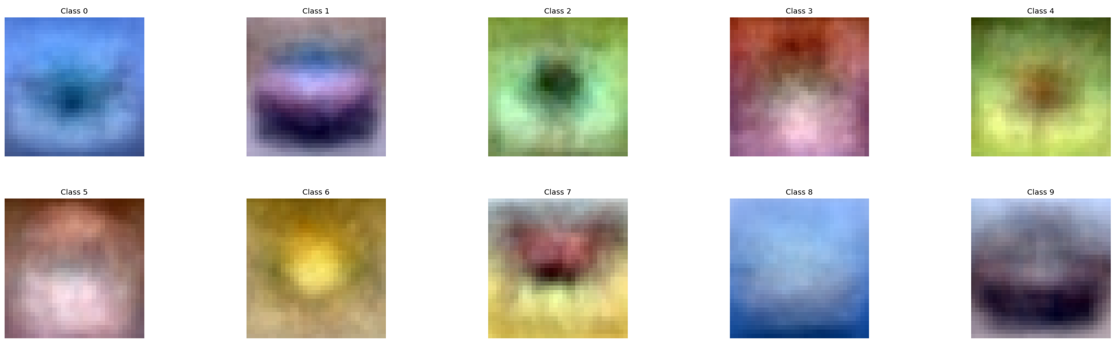



**Figure 3: Visualized weight matrices for classes**
**1st row (left to right): Airplane, automobile, bird, cat, deer**
**2nd row (left to right): Dog, frog, house, ship, truck**

Levent Güner
leventg@kth.se

Figure 1 shows how accuracies change over epochs. Bigger learning rate values yield faster results. However, if it is too high, it gives us inaccurate results.

Figure 2 shows how losses over epoch are changing. While it is expected to see the loss to decrease over epochs, it is inaccurate while learning rate is 0.1.

Figure 3 shows the weight matrices after training processes.

**Conclusion and Discussions**

It is obvious in the results that optimization of learning rate is so important. Learning rate determines the speed of training process. Therefore, higher learning rates yield faster results. However, if it is too high, the model gives us inaccurate results. As we can see in this model, learning rate 0.1 is too high for this case. So that the model gave us inconsistent results.

One other key factor is L2 regularization, which is determined by λ. The formula of L2 regularization is:

$$\lambda \sum_{i,j} {W_{ij}}^2$$

λ, the L2 coefficient determines the regularization. The aim of regularization is to prevent the model from overfitting. It is added to cost function and changes directly. While low values of λ is causing overfitting, higher values may cause underfitting. Therefore, like learning rate, optimizing λ value is so important in general. However, the dataset was quite small and does not require regularization. Also the results are proving the statement.

Despite being a single layer neural network, the model gave not so bad results. Visualization of weight matrices shows us how the model is trained and reached fitted results. Also changing the optimization values showed how optimization is important in a network. Adding more layers will definitely increase the results.