DD2424 Deep Learning in Data Science

Report on Assignment 1

Styliani Katsarou
stykat@kth.se

March 4, 2020

1 Introduction

In this assignment an one layer network that classifies images is trained and tested, using mini-batch gradient descent.

2 Methods

The network is implemented from scratch using Python 3.7. The only libraries used are Numpy, Matplotlib and for the preprocessing of the data Pickle was used. The chosen dataset is CIFAR-10 which has 10,000 images of 10 classes.

In order to verify the correctness of the analytically computed gradients, the results obtained by the functions made are compared against numerical estimations of the gradients.

Different combinations of parameter settings are tested in order to explore the effect of changing the regularization term and learning rate on the network's achieved accuracy.

The results are presented in Section 3 and discussed in Section 4.

3 Results

3.1 Checking the Gradients

The analytically computed gradients were compared against two different versions of numerically computed gradients: the first is based on the centered difference formula and the second on the finite difference method. The aforementioned comparison was conducted using the relative error between the numerically computed gradients and the analytically computed gradients. The results of the comparison between analytically computed gradients and the ones computed numerically using the finite difference method can be visually inspected in Figure1.

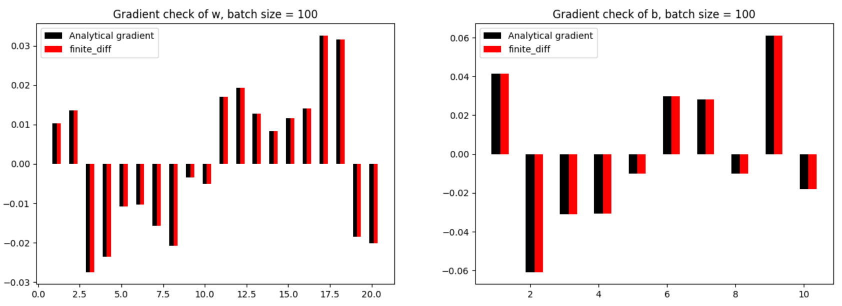

Figure 1: *Left:*Weight comparison between analytically computed gradients(black) and the ones computed numerically(red) using the finite difference method. *Right:*Bias comparison between analytically computed gradients(black) and the ones computed numerically(red) using the finite difference method

3.2 Experiments conducted using Different Parameter settings

As required by the assignment instructions, the implemented network's performance was examined by plotting the total cost and accuracy after each epoch. This procedure was repeated for two different learning rate values, 0.01 and 0.1m, for different regularization term values and the plots can be seen in Figure 2.

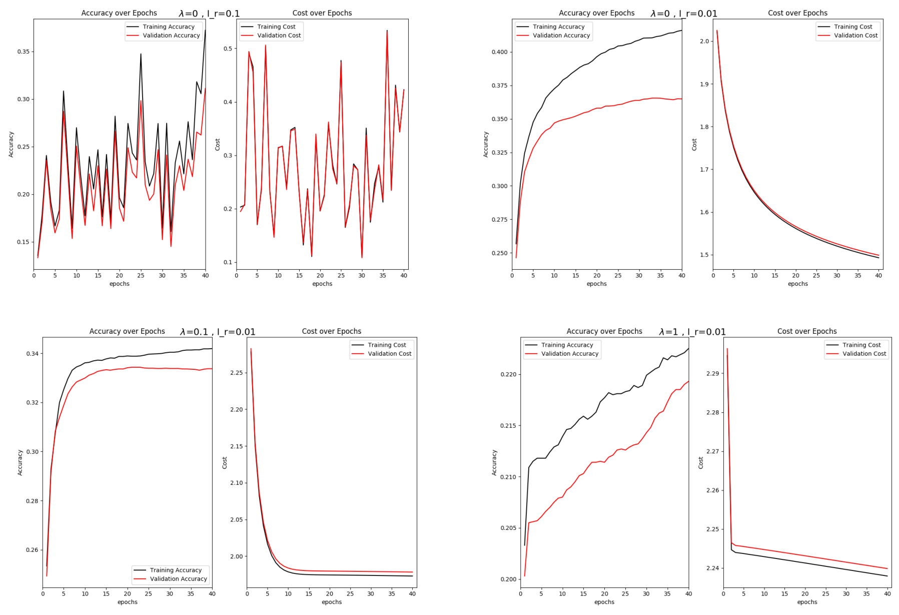

Figure 2: (left)

The combination of learning rate and regularization term parameter values that were tested, along with the corresponding train and test accuracy values obtained, are shown in the following table:

Results for different parameter settings		
Experiment settings	Train Accuracy	Validation Accuracy
$\lambda = 0$ & $lr = 0.1$	0.42	0.37
$\lambda = 0$ & $lr = 0.01$	0.37	0.22
$\lambda = 0.1$ & $lr = 0.01$	0.34	0.33
$\lambda = 1$ & $lr = 0.01$	0.22	0.22

3.3 Weight matrix visualization

Figure 3 shows how the visualization of the weight matrix of the first class can vary based on the different parameter setting scenarios and Figure 4 shows the visualization of the weight matrices of all classes for learning rate equal to 0.01 and regularization parameter equal to zero.

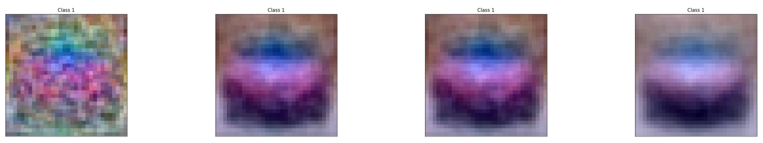

Figure 3: Visualization of the weight matrix of the first class for the following parameter settings: (i) $\lambda = 0$ & $lr = 0.1$ (ii)$\lambda = 0$ & $lr = 0.01$(iii) $\lambda = 0.1$ & $lr = 0.01$(iv)$\lambda = 1$ & $lr = 0.01$

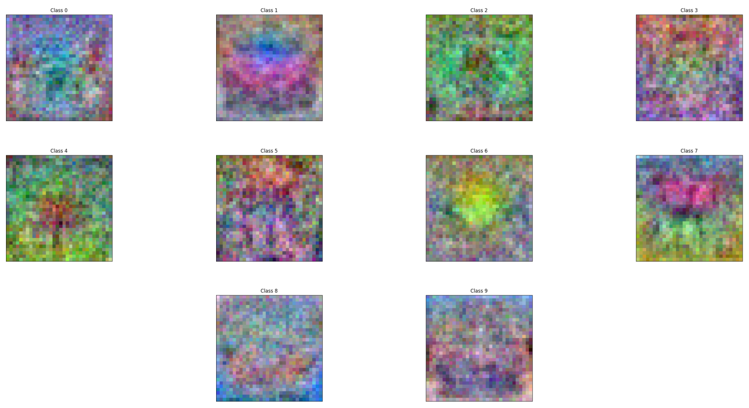

Figure 4: Visualization of the weight matrices of all classes for $\lambda = 0.1$ & $lr = 0.01$

4 Discussion of Results

4.1 Checking the Gradients

As can be seen in Figure 1, the numerically computed gradients are close enough to the analytically computes ones, using the the finite difference method. Thus, the numerically computes gradients can be approved for being used for the rest of this assignment.

4.2 Experiments conducted using Different Parameter settings

4.2.1 Learning Rate

For this assignment, two different leaning rate values were put to the test: 0.1 and 0.01. A quick look at Figure 2 leads to the conclusion that 0.1 is too large of a value for the network to learn: there is an intense oscillating behavior of the learning rate, and this very unstable training process can be interpreted as inability to learn. This is validated by the plot of the Cost, which is not steadily decreasing, instead it operates on the same oscillating pattern principle.

Decreasing the learning rate down to 0.01 leads to an important improvement of the network's ability to learn. The plots now vary depending on the different regularization term values chosen.

Even smaller learning rates were tested (10^{-6}) and they rendered the network unable to learn, since it was then stuck in a too long learning process.

4.2.2 Regularization Parameter

On a basis of keeping the learning rate constantly equal to 0.01, three different regularization parameter values were tested. A careful inspection of Figure 2 reveals that no $\lambda = 0$ results in validation accuracy much lower than training accuracy and the traning and validation cost reduction seems to need more epochs in order to converge to a steady value. Full regularization ($\lambda = 1$) results in a half the accuracy achieved for $\lambda = 0$ and a general poorer performace of learning. When $\lambda = 0.1$ the training accuracy is higher than the validation accuracy but not far above it, and the cost reduction indicates convergence of the weights learnt.

Since regularization is the penalty that refrains the weights from growing too large, and taking into account the aforementoned observations one could claim that $\lambda = 0$ leads to overfitting, $\lambda = 1$ resembles an underfitting behavior and $\lambda = 0.1$ seems to be the ultimate choice for λ out of the three.

As a general conclusion derived after these experiments, the parameter settings of $\lambda = 0.1$ and learning rate $= 0.01$ are the optimum choices for these hyperparameters.

4.3 Weight matrix visualization

The weight matrix visualtization of class 1 depicted in Figure 3, is in compliance with the conclusions derived for the various learning rate and regularization terms mentioned in Section 4.2.1 and4.2.2. When learning rate is equal to 0.1 the model cannot learn the weights well enough leading to the worst outcome out of the four. A wise choice of the learning rate leads to nicer weight matrix visualizations, but then the extreme case of applying full regularization leads to worse results. The images for $\lambda = 0$ and $\lambda = 0.1$ seem much alike, mainly because the two values are close enough. Figure 4 shows the weight matrices of all classes for the optimum choice of parameters, that is $\lambda = 0.1$ and learning rate $= 0.01$.