Deep Learning in Data Science

Report assignment 2

Two layer Neural Network on Cifar-10

Tobias Hoeppe

April 19th, 2021

1 Introduction

In this report we are going to investigate the influence of cyclic learning and the regularization parameter on the performance on a two-layer Neural Network with 50 hidden nodes.The Networks will be trained on 32 x 32 images from the Cifar-10 data-set.

2 Model configuration and Method

For the hidden layer the ReLu is used for activation and for the output layer Softmax. The loss is calculated by cross entropy (we will refer to the loss with the regularization term as cost). To ensure the correctness of the gradients of the weight matrix W and the bias b, I compared the analytical computed gradients with gradients computed by the temporal difference methods. The results with different Hyper-parameters are shown in Table 1 .

	relative error W_1	relative error b_1	relative error W_2	relative error b_2
$\lambda = 0$	2.175e-09	7.7e-11	1.77e-10	5.7e-11
$\lambda = 0.001$	5.7163e-08	7.7e-11	2.427e-08	5.7e-11
$\lambda = 0.1$	9.088e-09	2.38e-10	2.869e-09	1.66e-10

Table 1: Relative error between numerical and analytical computed gradients

3 Experiments

The experiments are all conducted with cyclic learning [1]. The notation is taken from the assignment itself. Also, we will not specify the number of epochs e as a Hyper-parameter, since it is determined by the step size ns, the number of

cycles trained nc, the batch size bs and the number of training samples n.

$$e = 2\frac{bs * ns * nc}{n} \tag{1}$$

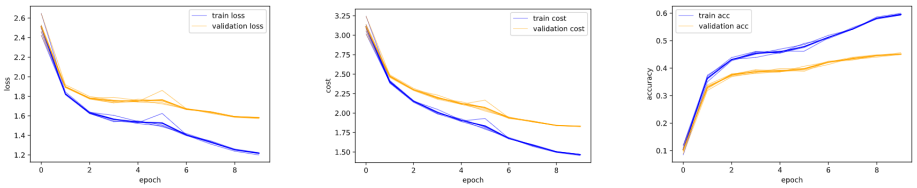
In the experiments we will only vary the step size ns, the number of cycles nc and $lambda$. The other Hyper-parameters will stay the same and can be found in Table 2.

batch size (bs)	η_{min}	η_{max}
100	10^{-5}	10^{-1}

Table 2: Hyper-parameters

3.1 Cyclic learning

First we are going to replicate the results given in the assignment for cyclic learning. For both experiments we chose the regularization parameter $\lambda = 0.01$ In the first experiment we have trained 5 Networks on the above mentioned parameters and with $ns = 500$, $nc = 1$. We can see a small oscillation (peak) when the learning rate increases. Also, the variance seems to be a bit higher at this point. The curves seem less smooth compared to performance curves with constant learning rate, as we experience rapid changes in the slope each time the learning rate gets updated (see Figure 1).



(a) Loss history of the Network

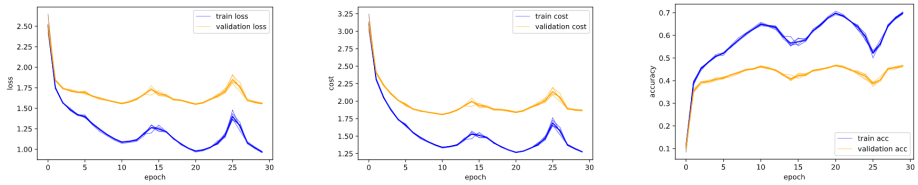
(b) Cost history of the Network

(c) Accuracy history of the Network

Figure 1: History of the performance

In the second Experiment the Hyper-parameter setting is the same as before only that we change the number of cycles trained $nc = 3$ and the parameter $ns = 800$. Now, the oscillation is clearly visible in the performance plot Figure 2 . The Model actually does visit several minima and again we can see higher variance during the time when the model is training with a high learning rate.

Finally we compare the performance on the test set vial the accuracy in Table 3. We can see that the second Model does perform better on all three sets. This is not really surprising, as it does train longer.



(a) Loss history of the Network

(b) Cost history of the Network

(c) Accuracy history of the Network

Figure 2: History of the performance

	train accuracy	validation accuracy	test accuracy
$ns = 500$, $nc = 1$	0.59542 $\pm$0.00266	0.45124 $\pm$0.00242	0.46086 $\pm$0.0015
$ns = 800$, $nc = 3$	**0.69904** $\pm$0.00385	**0.46518** $\pm$0.00324	**0.4724** $\pm$0.00304

Table 3: Final accuracy performance

3.1.1 Conclusion

We can see that the Network does visit a minima when the learning rate decreases and diverges out of one with increasing learning rate. As the minima have different values, one can assume that the Network actually visits different minima during training. The higher variance on the different peaks might be due to the fact, that training is less stable, when the learning rate is high.

3.2 Lambda Search

In this experiment we did search for an optimal regularization parameter λ. To ensure that the tests are valid, the Networks had to be train on most of the data, since an increase in training data has a regularizing effect. Due to this, each Network could only be trained once and therefore no statistics about the variance are available. The Hyper-parameters where chosen as in the experiments above except that we varied ns and nc
In the first experiment we trained on 4500 data points. And according to [1], we did set $ns = 2\lfloor \frac{n}{bs} \rfloor$ and $nc = 2$. Nine exponents where generated on a uniform distributed grid in the interval $[-5, -1]$ and the regularization term was set to λ^k $k \in [-5, -1]$. In Table 4 we can see that the best values where achieved by $\lambda \in$ [1e-3,1e-4]. Also, $\lambda = 1e - 5$ performs well, but when we look at the difference between train (61 %) and validation loss (50.2 %), we can see that we over-fit extremely and that is why we do not conduct further searches on that area.

λ	0.1	0.03594	0.01292	0.00464	0.00167	0.0006	8e-05	3e-05	1e-05
validation accuracy	0.385	0.459	0.487	0.503	0.5	**0.506**	0.495	0.497	0.502

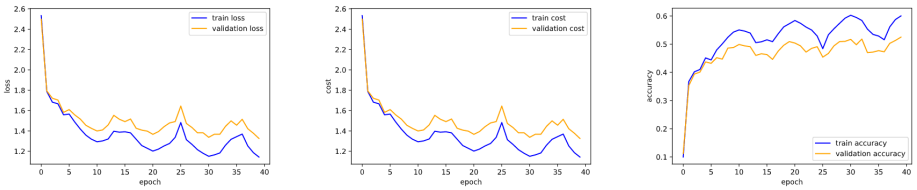
Table 4: The results from the first training on different λ

Now, given this results we did narrow our search to values for $\lambda \in$ [1e-2,1e-5] (the Interval was chosen a bit bigger to ensure we do not miss important values). In this Interval we did generate eight λ and trained for three cycles. In Table 5 we can see the results.

λ	1e-05	3e-05	7e-05	0.00019	0.00052	0.00139	0.00373	0.01
validation accuracy	0.5146	0.5158	0.5142	0.5176	0.516	**0.5206**	0.5182	0.5064

Table 5: The results from the narrow search

The best results on the validation set seems to be achieved by $\lambda = 0.00139$. Therefore this λ was chosen to be trained on almost the entire data-set (only 1000 samples where left out for validation) and 4 cycles. The development of the performance measures can be seen in 3 and the final accuracy performances in Table 6 .



(a) Loss history of the Network

(b) Cost history of the Network

(c) Accuracy history of the Network

Figure 3: History of the performance measures given the model with $\lambda = 0.00139$

train accuracy	validation accuracy	test accuracy
0.60016	0.525	0.5215

Table 6: Final accuracies of the Network with $\lambda = 0.00139$

3.2.1 Conclusion

In the history of the performance measures Figure 3 we can see that there is almost no over-fitting. Given this and the accuracy performance see Table 6 we can conclude, that $\lambda = 0.00139$ seems to be a good choice for our Model.
That the optimal λ is quite low, can be explained by the fact, that we train a

rather simple model on quite a large amount of data. This does have a good regularizing effect itself and therefore a higher λ leads to over-regularization. This is also the reason why we still see decent performance with $\lambda = 1e - 5$. Also, in Figure 3 we can see the oscillations induced by the cyclic learning quite nicely, and the fact that we do reach different minima over time. Thus, we can confirm observations made in section 3.1.1,

References

[1] Leslie N. Smith. Cyclical learning rates for training neural networks, 2017.