ASSIGNMENT 3

Diego Argüello Ron

Deep Learning in Data Science (DD2424)

28 June 2018

1 Introduction

The objective of this assignment is to improve the results obtained in the first ones by adding more hidden layers to the Network as well as applying momentum and Batch Normalization.

2 Checking Gradients

Like in the previous assignments, making sure that the gradients are being computed correctly is of the utmost importance. Thus, the calculated gradients were compared with the ones calculated numerically by obtaining the relative error between them. Being g_n the value of numerically computed gradient and g_a the value of the analytically computed gradient, the relative error will be:

$$\frac{|g_a - g_n|}{max(esp, |g_a| + |g_n|)} \tag{1}$$

where esp is a very small positive number, taken as 0.

The relative error was calculated for mini-batches of size 10, 20 and 30 with no regularization. The network used was a 2-layer Network.The next results were obtained:

Batch size	Relative Error b_1	Relative Error b_2	Relative Error W_1	Relative Error W_2
10	1.2866e-07	1.0594e-10	5.8156e-07	1.2639e-09
20	1.4993e-07	2.0207e-10	7.6248e-07	1.4795e-09
30	2.6862e-07	3.0972e-10	1.3399e-06	2.1623e-09

As it can be seen in the results of the table, the gradients are being calculated properly.

3 Comparison of 3-layer Networks with and without Batch Normalization

In this section a 3-Layer Network has been trained using the parameters $\eta = 0.1$ and $\lambda = 1 \cdot 10^{-4}$. As it can be seen below, there is a huge increase in the performance thanks to adding Batch Normalization.

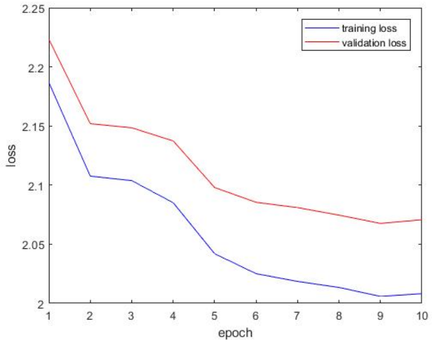

Figure 1: 3-Layer Network without Batch Normalization

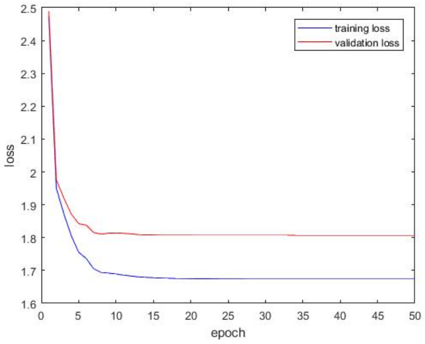

Figure 2: 3-Layer Network with Batch Normalization

4 Hyper-parameters Search

In this section, by setting boundaries to the values of λ and η and selecting the best values of the accuracy for the different pairs in these intervals, the boundaries are being reduced until an optimal pair of values is found.

Three intervals have been tested in oder to get the values of the hyper-parameters: $0.7 > \eta > 0.01$ and $0.1 > \lambda > 1 \cdot 10^{-6}$; $0.2 > \eta > 0.08$ and $3 \cdot 10^{-3} > \lambda > 4 \cdot 10^{-5}$; $0.09 > \eta > 0.07$ and $3 \cdot 10^{-4} > \lambda > 5 \cdot 10^{-5}$ .The pairs calculated were 50 because of the high time it took to calculate the values.

Thus the chosen values were at the end: $\eta = 0.0864$ and $\lambda = 1.8245 \cdot 10^{-4}$, achieving a train accuracy of 62.68% and a test accuracy of 45.02%

.

Here, the loss for the training and test data is displayed:

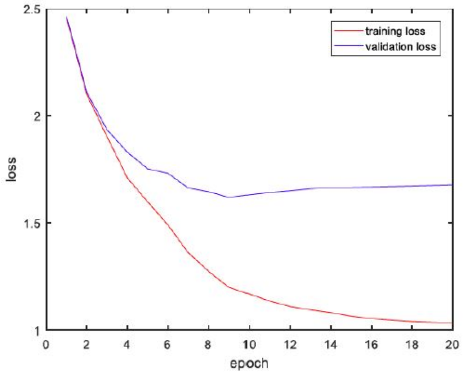

Figure 3: Loss functions for the best hyper-parameters

5 Study of the influence of η in a 2-layer Network with and without Batch Normalization

Finally, a last analysis was carried out. It consisted in training a 2-layer network, with batch normalization using 3 different learning rates (small, medium and high) for 10 epochs in order to see the influence of this parameter in the learning process.Thus, the next results were obtained:

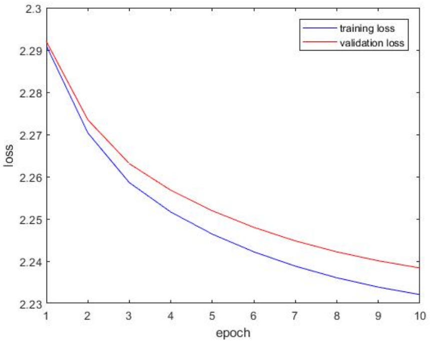

Figure 4: 2-Layer Network with Batch Normalization and $\eta = 0.001$

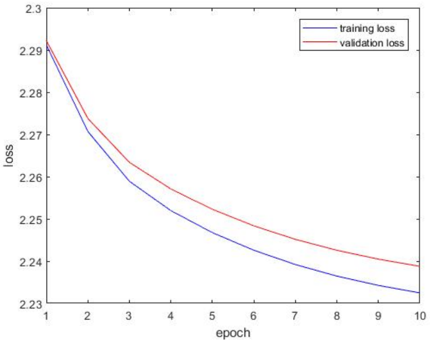

Figure 5: 2-Layer Network with no Batch Normalization and $\eta = 0.001$

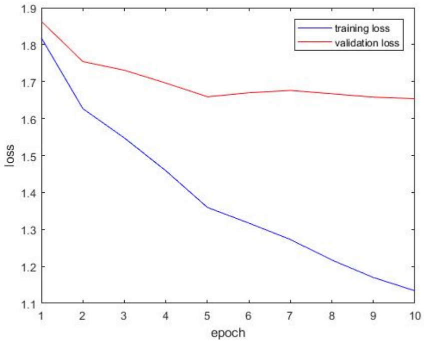

Figure 6: 2-Layer Network with Batch Normalization and $\eta = 0.09$

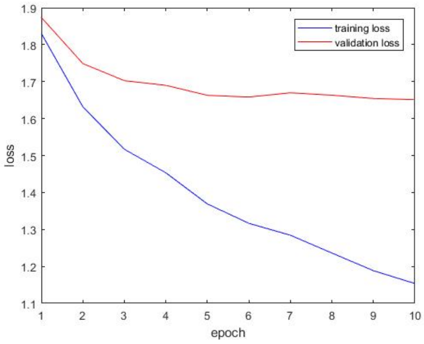

Figure 7: 2-Layer Network with no Batch Normalization and $\eta = 0.09$

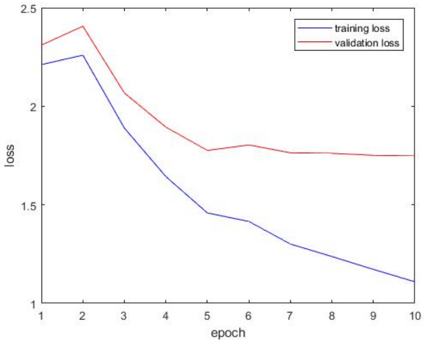

Figure 8: 2-Layer Network with Batch Normalization and $\eta = 0.2$

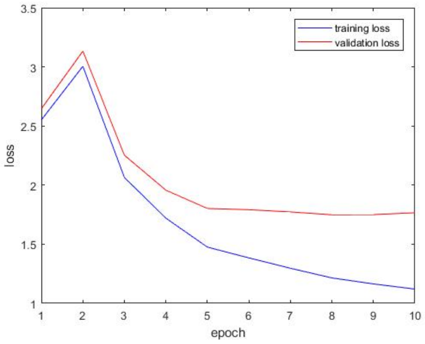

Figure 9: 2-Layer Network with no Batch Normalization and $\eta = 0.2$

As it can be seen, for a 2-Layer Neural Network adding Batch Normalization does not suppose such a high improvement in the performance of the network. Nevertheless, if the learning rates are small, the training loss barely change, meanwhile for large learning rates the learning is unstable.