Deep Learning in Data Science - Assignment 2

Fernando García Sanz

April 6, 2020

Abstract

The scope of this assignment consists in building and training a two layer network which has multiple outputs. This network is used to classify the images contained in the CIFAR-10 [1] dataset. The network is trained using mini-batch gradient descent over a cost function which computes the cross-entropy loss of the classifier applied to the labelled training data and an L_2 regularization term over the weight matrix.

1 Comparison of analytical and numerical gradients

To compare the obtained results with the analytical and numerical gradients, small batches have been used to allow computing the results in a reasonable amount of time. The performed tests are the following:

	Analytical - Numerical		Analytical - Slow	
Weight Gradient (W_1, W_2)	4.415e-08	1.944e-07	4.717e-14	3.997e-13
Bias Gradient (B_1, B_2)	4.415e-08	4.096e-07	7.745e-13	4.718e-18

Table 1: Difference between analytical and numerical calculations of gradients with generating bias and weights with seed 42 over the 20 first samples.

	Analytical - Numerical		Analytical - Slow	
Weight Gradient (W_1, W_2)	4.117e-08	1.757e-07	1.463e-14	1.021e-12
Bias Gradient (B_1, B_2)	4.114e-08	4.322e-07	2.431e-12	4.441e-12

Table 2: Difference between analytical and numerical calculations of gradients with generating bias and weights with seed 400 over the 20 first samples.

	Analytical - Numerical		Analytical - Slow	
Weight Gradient (W_1, W_2)	3.407e-08	1.683e-07	5.171e-14	4.885e-13
Bias Gradient (B_1, B_2)	4.288e-08	4.295e-07	9.150e-13	5.551e-19

Table 3: Difference between analytical and numerical calculations of gradients with generating bias and weights with seed 42 over the 20 random samples.

	Analytical - Numerical		Analytical - Slow	
Weight Gradient (W_1, W_2)	3.255e-08	1.552e-07	2.145e-14	2.220e-13
Bias Gradient (B_1, B_2)	4.251e-08	4.301e-07	7.851e-13	2.220e-12

Table 4: Difference between analytical and numerical calculations of gradients with generating bias and weights with seed 400 over the 20 random samples.

As it can be seen, the differences between the analytically calculated and the numerical ones are quite small, from the order of 10^{-7} as maximum. Moreover, if we take into account that the

slow calculation is more precise, and it is in this case where the differences are smaller, we can assume that the analytical calculation of the gradients is well performed.

2 Cyclical learning rates

The implementation of cyclical learning rates allows to improve the performance of the classifier. Two different configurations have been tested here:

• Configuration 1: η_{min} = 1e-5, η_{max} = 1e-1, $\lambda = 0.01$, $n_s = 500$.

This configuration is employed for one cycle of training. Being the training set composed by 10000 samples, and being the batch size equal to 100, it is necessary to perform 10 epochs to complete once cycle.

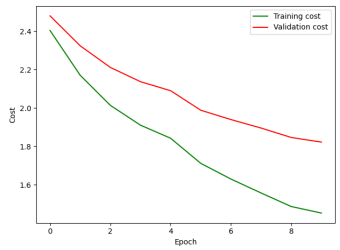

Figure 1: Cost over training epochs.

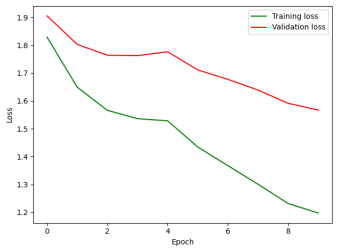

Figure 2: Loss over training epochs.

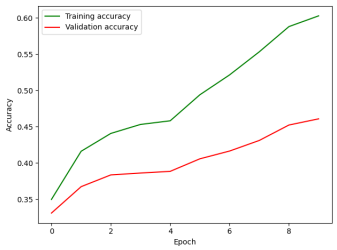

Figure 3: Accuracy over training epochs.

• Configuration 2: η_{min} = 1e-5, η_{max} = 1e-1, $\lambda = 0.01$, $n_s = 800$.

This configuration is employed for three cycles of training. Being the training set composed by 10000 samples, and being the batch size equal to 100, it is necessary to perform 48 epochs to complete the three cycles of training.

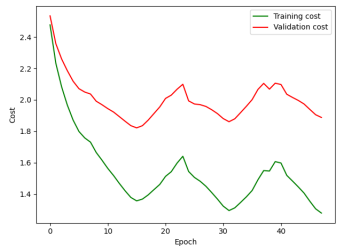

Figure 4: Cost over training epochs.

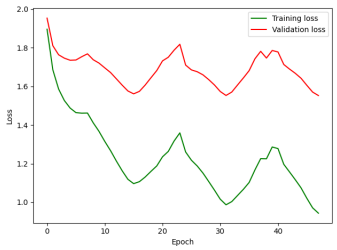

Figure 5: Loss over training epochs.

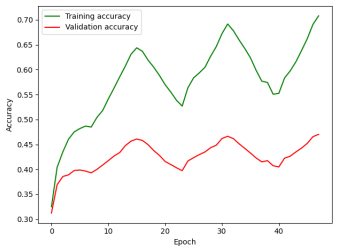

Figure 6: Accuracy over training epochs.

As can be seen, the difference when applying more cycles lies in the beak shape that can be observed in the plots of the second configuration. Due to the different cycles employed, in which the η value increases linearly from the minimum to the maximum value, to then decrease, it can be seen how loss, cost, and accuracy also vary in a triangular shape. The accuracy obtained over the test set with the first configuration is equal to 45.78%, while the one retrieved from the second configuration is equal to 46.74%, being able to observe a slight improvement due to the increment in the number of steps per cycle, as well as in the number of cycles performed.

3 Lambda values search

It is also necessary to find a good regularization parameter when training a model, which will vary due to its characteristics. To do so, a mechanism which employs both coarse and fine search has

been used. It is also necessary to remark that 45000 samples from the dataset have been used for training, while the 5000 remaining ones are used as the validation set.

3.1 Coarse search

To perform this search, it is necessary to set some interval in which looking for the lambda values. To accomplish this, the interval has been defined in $\log_{10}$ scale, being the boundaries [-5, -1].

During each search, a random value if obtained from that interval, kept fixed, and the network is trained, using always the same settings since otherwise the comparison of performances would not be fair.

The network has been trained eight times, using as parameters $n_{batch} = 100$, $\eta_{min} =$ 1e-5, $\eta_{max} =$ 1e-1 and $n_s = floor(N/n_{batch})$. With these parameters, the best lambdas can be selected according to the accuracy over the validation set. The three best ones were the followings, according to the retrieved accuracy:

- Best accuracies in validation: [0.5198, 0.5164, 0.5162]
- Best lambdas in validation ($\log_{10}$ scale): [-2.7488288, -2.99472691, -4.46580721]

3.2 Fine search

Once the best values of lambda are obtained, it is time to perform a more precise search. To do so, the interval of values is modified, employing as limits of it the best two values of lambda in logarithmic scale. Let's say our interval of values is know [λ_1, λ_2].

Now the same procedure is performed, employing eight different values obtained from the new interval, but also considering the previously three best found values of λ. Also, the number of cycles employed has been increased to four in order to perform a more exhaustive search. Employing this technique, it has been possible to obtain even better values of lambda:

- Improved accuracies in validation: [0.5268, 0.5248, 0.5244]
- Improved lambdas in validation ($\log_{10}$ scale): [-2.94991004, -2.88229322, -2.92952672]

As it can be seen, the accuracy values obtained outperform those obtained in the coarse search.

3.3 Best lambda training

Now that a good value of lambda is found, it is possible to perform a more exhaustive training using almost the entire dataset, saving only 1000 values for validation. Moreover, the network has been trained for 3 cycles and the value of n_s employed has been doubled.

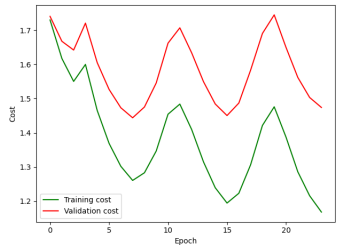

Figure 7: Cost over training epochs.

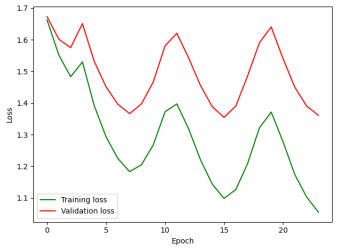

Figure 8: Loss over training epochs.

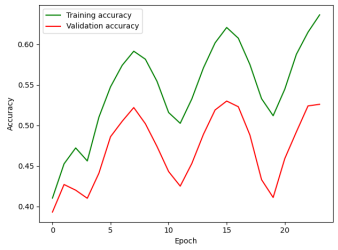

Figure 9: Accuracy over training epochs.

Training for a total of 24 epochs have retrieved substantial improvement in the performance of the network. The accuracy over the test data has been equal to 52.21%. As it can be seen, training with more samples and a good selection of lambda provides a substantial improvement respect to previous configurations.

References

[1] A. Krizhevsky. The cifar-10 dataset. `https://www.cs.toronto.edu/~kriz/cifar.html`. Accessed: 21-03-2020.

[2] L. N. Smith. Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 464–472. IEEE, 2017.