DD2424 Deep Learning in Data Science: Assignment 4

Léo Noharet - lnoharet@kth.se

May 18, 2022

The steps to check that the gradient computations are correct are very similar to the previous assignments. I firstly translated the matlab code provided for the numeric computation to python code. Then, I computed the relative error between analytical and numeric computations for all gradients, with eps set to 10^{-6}. I also compared the gradients computed numerically and analytically with the help of the assert_almost_equal(actual, desired, decimal) function from the numpy.testing python library. This function verifies that the elements of the matrices *actual* and *desired* satisfy the following: $abs(desired - actual) < 1.5 * 10^{-decimal}$. [Dev] I used this with the parameter decimal set to 8, which proves that the difference is small between the gradient matrices. Both analytical and numerical gradients were computed on one sequence length of the data, i.e. the first 25 characters as input data and characters 1-26 as label data and with the amount of nodes in hidden layer = 100. The differences are shown in figure 1.

Grad	Diff
b	1.8481122250479325e-09
c	3.216327131607271e-10
U	3.571355731701593e-09
W	1.0578422041798475e-07
V	9.568499888414626e-09

Figure 1: Difference between numerical and analytical gradient computations for each parameter of the model

1 Graph of the smooth loss function for a longish training run

The smooth loss function is shown in figure 2. The training was performed during 10 epochs with hyper parameters: lr = 0.01, sequence length = 25, sigma = 0.1, amount of nodes in hidden layer =

100.

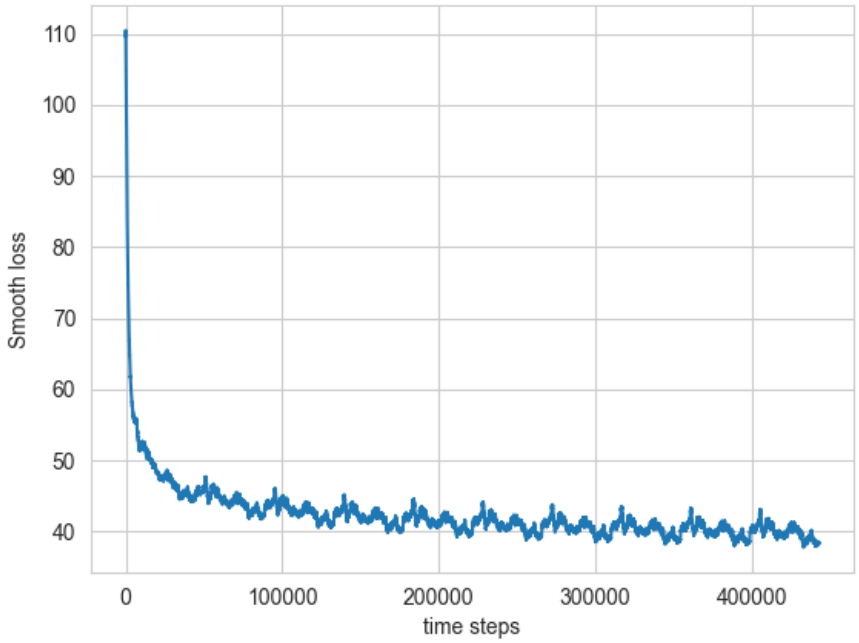

Figure 2: Smooth loss function of the model

2 Evolution of synthesized text

The output of the synthesized text every 10,000th time step for 100,000 time steps can be seen in figure 3

3 Passage from best model

After training the model a few times, the smallest loss achieved was approximately 37.8 after 433289 time steps or 9 epochs of training. The hyper-parameter setting used was the same as for all other training for this assignment: lr = 0.01, sequence length = 25, sigma = 0.1, amount of nodes in hidden layer = 100. The 1000 characters generated with this model can be seen in figure 4

Figure 3: Synthesized text every 10,000th step for 100,000 steps.

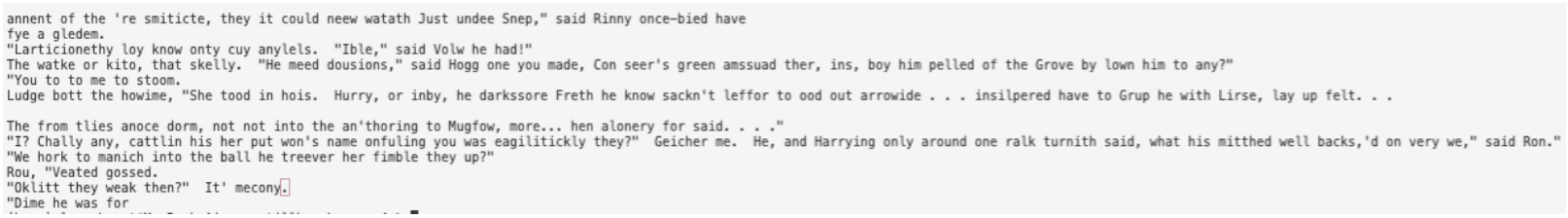

Figure 4: Synthesized text of best model

References

[Dev] Numpy Developers. numpy.testing.assert_almost_equal. `https://numpy.org/doc/stable/reference/generated/numpy.testing.assert_almost_equal.html`. Accessed: 2022-04-04.