Report assignment 4

Shiva Besharat Pour
shivabp@kth.se

Introduction

This assignment focused on training a Recurrent Neural Network (RNN in short) in order to synthesise parts of text from the book of Goblet of fire.

Implementation and results

Gradient evaluation

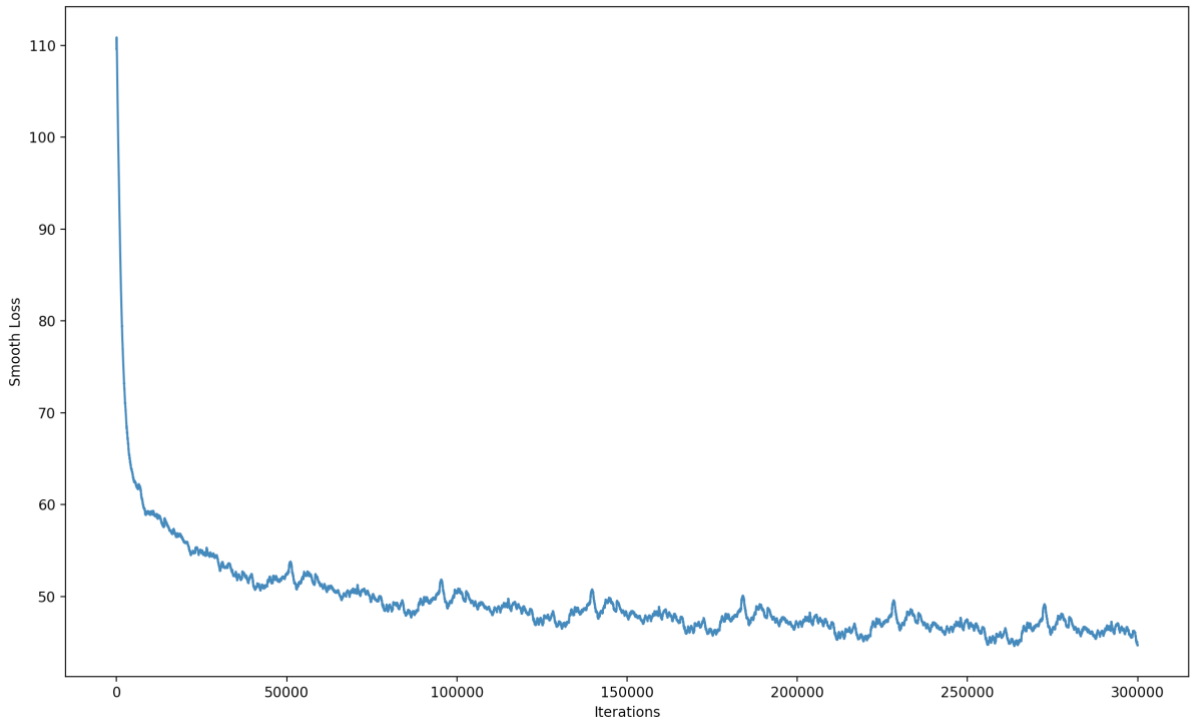
I implemented the analytic forward pass based on the lecture slides of lecture 9. In order to reassure myself the computation of the forward pass is correct, I used the given MATLAB code in order to implement the numerical version of the forward pass that is supposed to be more precise. The parameter values I used for the comparison are as recommended in the instructions PDF. The results are as follows:

```
Shivas-MacBook-Pro:Lab 4 shivabp$ python3 solution.py
gradb results:
Average of absolute differences is:  1.383204954394368e-07

gradc results:
Average of absolute differences is:  1.5429177935288168e-05

gradW results:
Average of absolute differences is:  5.698006805081291e-09

gradU results:
Average of absolute differences is:  2.1557255654903075e-09

gradV results:
Average of absolute differences is:  1.5489831639444823e-09
```

As the difference between the numeric and analytic forward pass result for each of the weight and bias vectors is small, I concluded that my analytic forward pass seems to be reliable.

Evolution of Smooth Loss:

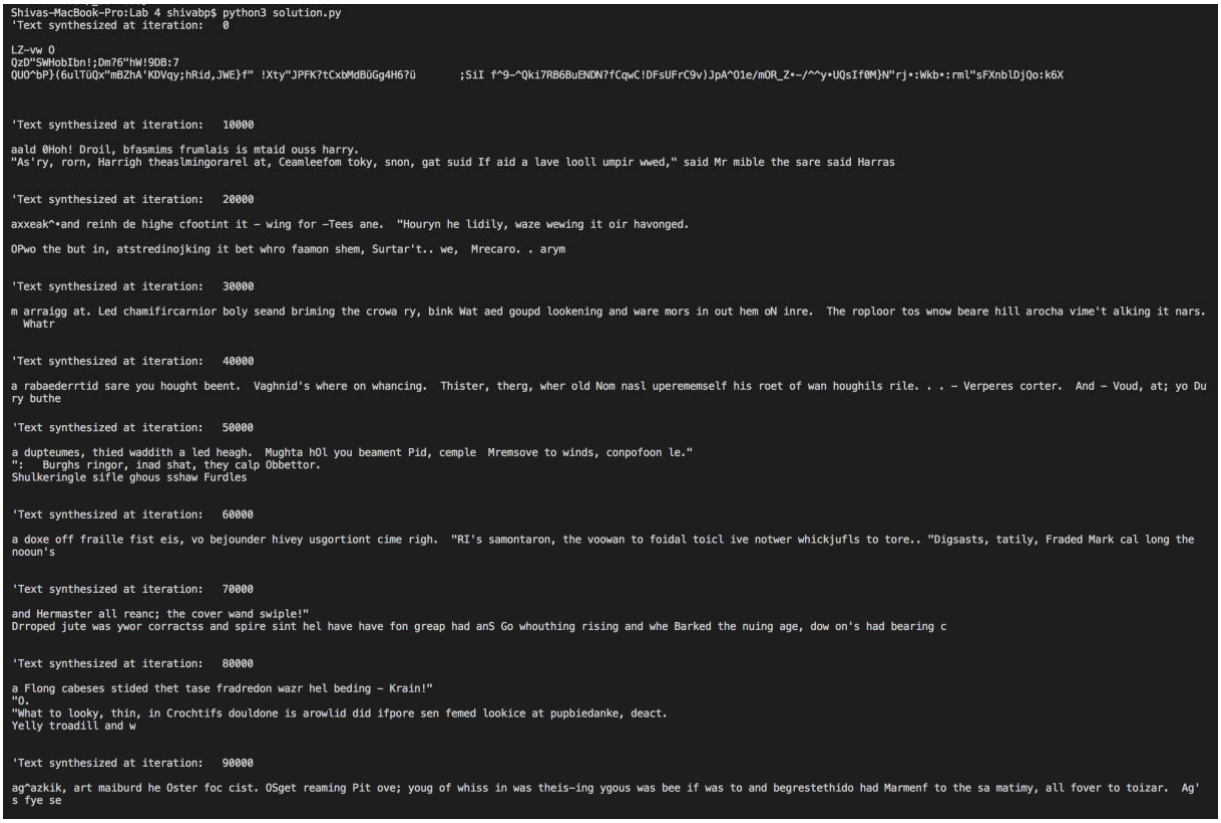
Once the correctness of my forward pass was confirmed, the network was trained for 300,000 iterations, with 200 characters long samples. The reason I chose 300,000 iterations was because the screenshot provided on page 7 of the assignment also ends around 330,000 iterations. I thus wanted to more easily be able to compare my results and see if I am getting what I should be getting. The plot of the smooth loss is as follows:



The plot confirmed that my results seem to be as they should.

Sample text synthesis:

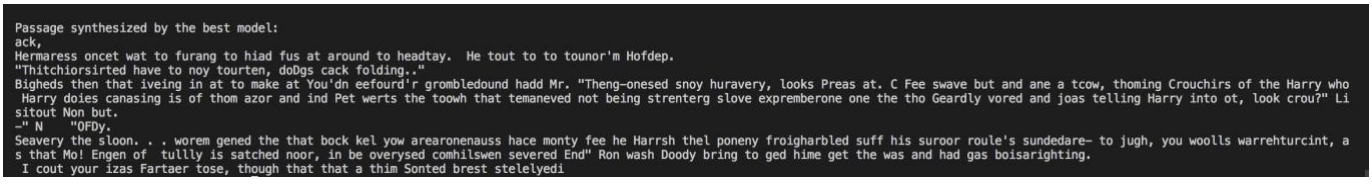
I trained the network separately for 100,000 iterations again with 200 characters long samples. At every 10,000th iteration, I did a synthesis of a sample text and the screenshots below show the results of all synthesised text throughout the training.



According to these results, the results of synthesis gets better and better over time. The synthesised output seems to be more realistic and the more character sequences correspond to actual words which is as expected.

Best model:

I used the training above in order to store the weight and bias vectors that resulted in the lowest smooth loss. I then used them as input parameters to synthesise a sample text of 1000 characters long as suggested by the assignment instructions. The synthesised output is shown below.



Not completely perfect but there are many recognizable words and even character sequences that are not actual words, are quite close to words that would be relevant.