KTH Royal Institute of Technology

DD2424 Deep Learning in Data Science

Assignment 3 Bonus Report

Improve the performance of RNN

Written by

Yutong Jiang

yutongj@kth.se

Date: May 22, 2022

1 introduction

In the first part, I train the model with Adam instead of AdaGrad to see if I could reach lower smooth loss score.

In the second part, I train with sequences from random locations in the text at each update iteration. Besides, I try the compromise solution to split the book into L chunks and then randomly choose the order of these chunks.

In the third part, I train increase the batch size to see if it could be used to speed up convergence.

2 AdaGrad

In this part, I train the model with Adam. The update equations for Adam could be shown as follows.

$$m^{t+1} = \beta_1 m^t + (1-\beta_1) * g_t \tag{1}$$

$$v^{t+1} = \beta_2 v^t + (1-\beta_2) * g_t . * g_t \tag{2}$$

$$\hat{m}^{t+1} = \frac{m^{t+1}}{1-\beta_1^t} \tag{3}$$

$$\hat{v}^{t+1} = \frac{v^{t+1}}{1-\beta_2^t} \tag{4}$$

$$x^{t+1} = x^t - \frac{\eta}{\sqrt{\hat{v}^{t+1}} + \epsilon} \hat{m}^{t+1} \tag{5}$$

The eta I choose is 0.001 and the performance of Adam and AdaGrad could be shown as follows.

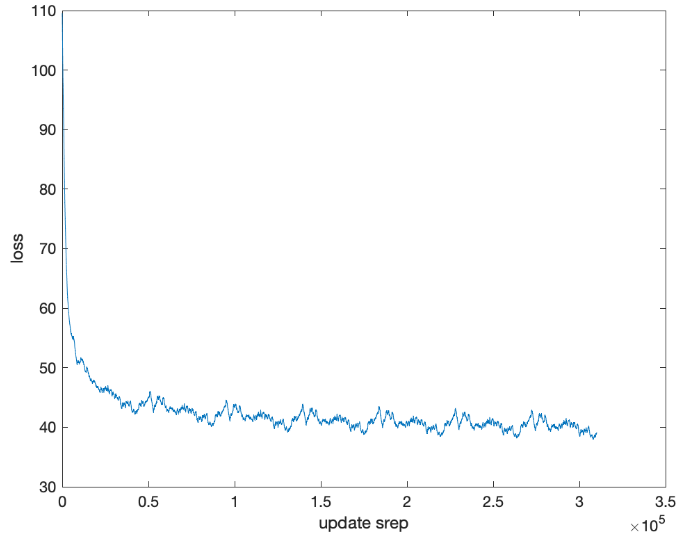

Figure 1: Figure of smooth loss of Adam when eta equals to 0.001

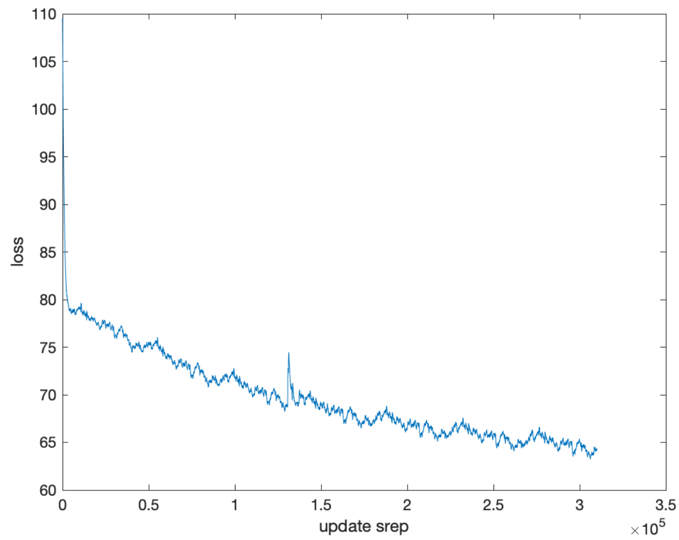

Figure 2: Figure of smooth loss of AdaGrad when eta equals to 0.001

From the two figures, we could find that when eta equals to 0.001, train model with Adam could converge in a faster way and its loss will be lower than that trained with AdaGrad.The smooth loss I could achieve with Adam is 38.7862.

3 sequences from random locations and a compromise solution of chunks

In this section, I choose sequences from locations as input and performance of smooth loss I could achieve could be shown as follows. The parameter is the same as that I set in basic part.

From the figure of smooth plot, we could find that the smooth loss I could achieve is 44.2908, which is pretty close to the smooth loss I could achieve with AdaGrad. The conclusion is that there is not much improvement by randomizing the input sequences when comparing with the smooth loss I could achieve in the basic part.

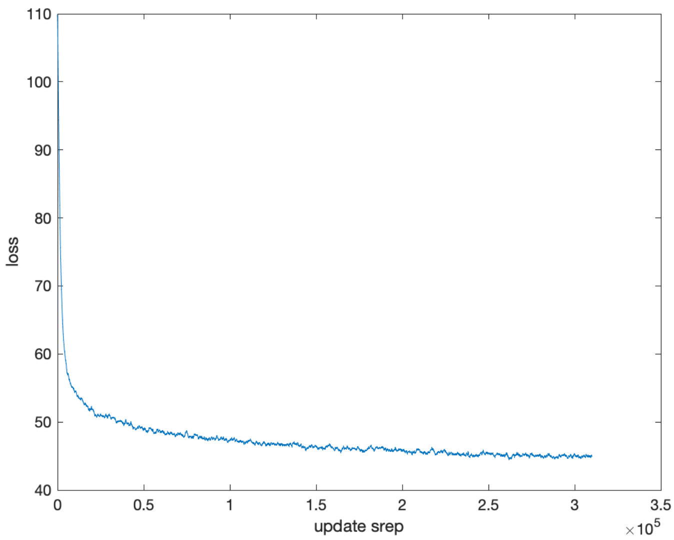

Figure 3: Figure of smooth loss trained with sequence from random locations

After that, I intend to explore the model using the compromise solution to split the dataset into several chunks. The dataset is split to 4 parts and for each epoch, it is visited randomly. The other parameter is set the same as before. The figure of smooth I could achieve could be shown as follows.

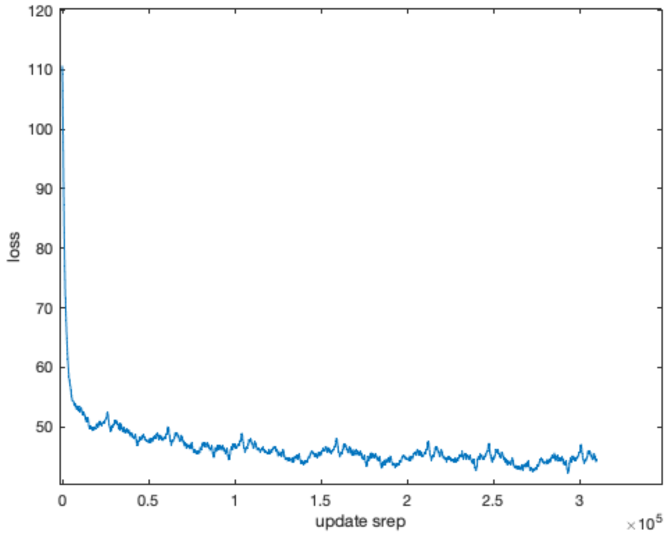

Figure 4: Figure of smooth loss trained with sequence from random chunks

The smooth loss is 42.2365, which is smaller than that I achieved without randomization. However, the convergence has not been speed up according to my results.

4 increasing the batch size

In this part, I try to change the batch size from 1 to 2 and keep the update steps unchanged. The figure of smooth loss I could achieve is as follows.

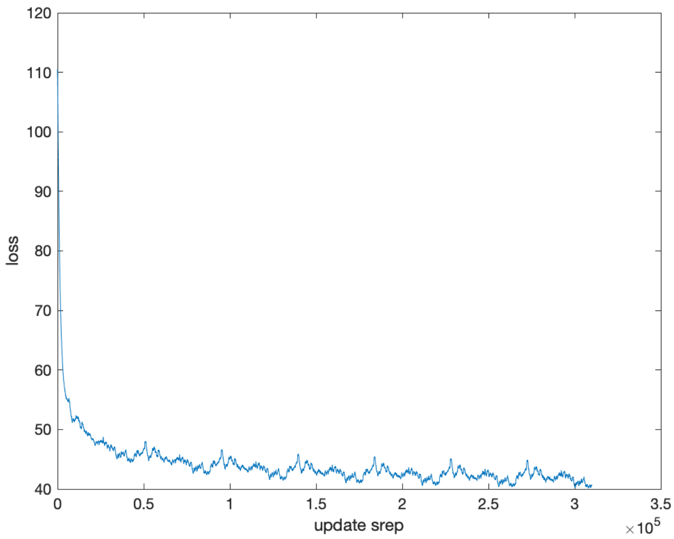

Figure 5: Figure of smooth loss trained by setting batch size to 2

We could find by changing the batch size, it could speed up the convergence. Besides, it could also achieve a lower smooth loss, which is 40.6888 when comparing that in basic part.