Deep Learning in Data Science
DD2424

Solution to Assignment 4

Yiming Fan
yimingf@kth.se

1 Introduction

This assignment aims at training a Recurrent Neural Network (RNN) using *gradient descent* method. The dataset used in this assignment is the complete book of *Harry Potter and the Goblet of Fire* by *J. K. Rowling*.

2 Methods & Mechanisms

2.1 Gradient Descent

Similar in the previous assignments this network contains a forward-pass and a backward-pass. The formulas for forward-pass are described in the lecture slides. The gradient descent formulas are:

$$\frac{\partial L}{\partial V} = \sum_{t=1}^{\tau} \mathbf{g}_t^T \mathbf{h}_t^T$$
$$\mathbf{g}_t = \frac{\partial L}{\partial \mathbf{o}_t} = (\mathbf{y} - \mathbf{p})^T$$
$$\frac{\partial L}{\partial \mathbf{c}} = \sum_{t=1}^{\tau} \mathbf{g}_t^T$$

and

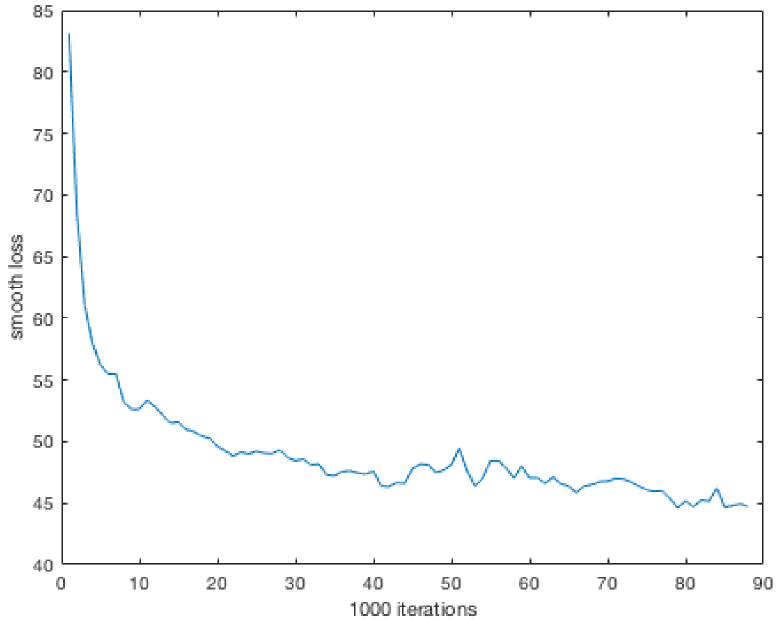
$$\frac{\partial L}{\partial W} = \sum_{t=1}^{\tau} \mathbf{g}_t^T \mathbf{h}_{t-1}^T$$
$$\frac{\partial L}{\partial U} = \sum_{t=1}^{\tau} \mathbf{g}_t^T \mathbf{x}_t^T$$
$$\mathbf{g}_t = \frac{\partial L}{\partial \mathbf{a}_t}$$
$$\frac{\partial L}{\partial \mathbf{b}} = \sum_{t=1}^{\tau} \mathbf{g}_t^T$$

The formulas above were proven correct by verifying through `ComputeGradsNum.m` provided in *KTH Social*. Setting the number of hidden states $m = 100$ the differences between calculated gradients and the numerical solutions do not exceed: 10^{-5} for $\frac{\partial L}{\partial \mathbf{c}}$, 10^{-7} for $\frac{\partial L}{\partial V}$, 10^{-6} for $\frac{\partial L}{\partial \mathbf{b}}$, 10^{-7} for $\frac{\partial L}{\partial U}$ and 10^{-9} for $\frac{\partial L}{\partial W}$.

2.2 Smooth loss function plot

2.3 Evolution of the training result

```
foo =
```




```
       10000

chars =

ndaiderttirk tf tasg  yn tose  aoreahe  th teat   "hrer nlan the r toom    aooked an tf


foo =

       20000

chars =

HIe waid He s hte  ahe wneeweaf tes sas   and hhe  ai hose  tote     . .a wor tndook tf


foo =

       30000

chars =

 haue shenk towe taf treoled anrund the r aoak    he  aaet d tn tarry sn ti wtarptrrt  t
```

```
foo =

       40000

chars =

g the r tasd  t "he wrtd  ttheeet toudedeeng tarry tnd toldedort ateettir d  Vheugh the


foo =

       50000

chars =

t   "ean  ahe    sa snd d tor  e  aoas ng the sas ah nrd the soar d  tvciende  ahanh aas


foo =

       60000

chars =

owd  aoom the sreugd   anpn  the   the serddn oney
 Hheu  trt  r d tneund the sruledhtf trnt ttaok teck to toatshe  aarte ahs heng txse  y


foo =

       70000

chars =

the sart  and ahietrenr tuelrkun snd Honedeaumend aoort ng tadl oomde rld tn tnsewe es


foo =

       80000

chars =

wo tte   aet tndutdrtageanttae has wooesey tn the si rht tf tes wuideenhh  eas ng torhed
```

```
foo =

       90000

chars =

t ht ttrteen ng a t  er hnl  ae wid af y horld tft thet hnmkns aas aas hotden er thanher


foo =

      100000

chars =

Iuawauketh tvdere ahrld ’t tn " Hut t wty rt the  wanht tasd tt e re tfd r  .  .umdinsn
```

2.4 Best model

We tried $m = 100$, $\eta = 0.1$ and trained the network with 660000 iterations. The smooth loss was 38.2217 eventually and the generated words are:

```
aP   CaTTER F T CoERToTTo- RnTMo
TIaEPTER TNE TTTHE To  ’ENVaGTt "T he wosyote  .wf togtle tardre nrswhill borled tn waho
 Ht waapk an t fasl tner ywneng toe cosyote  ahme tf tt. tatkews weur  d  ahgl  wogteng
 sarlotngaptlre wnau ahme hing ttaaigedtnd sewriele ted hevpentd the e  ahme hing toet w
```