CSC321: Programming Assignment 3

Due on Wednesday, March 22, 2017

Xiaodi Lu

March 23, 2017

Part 1

I used the sample function from $min - char - rnn.py$ added an alpha in the parameter and in the function I multiply it to the value assigning to y.
The output generated using different termperatures.
Below is the orginal function:

```
def sample(h, seed_ix, n):
  """
  sample a sequence of integers from the model
  h is memory state, seed_ix is seed letter for first time step
  """
  x = np.zeros((vocab_size, 1))
  x[seed_ix] = 1
  ixes = []
  for t in xrange(n):
    h = np.tanh(np.dot(Wxh, x) + np.dot(Whh, h) + bh)
    y = np.dot(Why, h) + by
    p = np.exp(y) / np.sum(np.exp(y))
    ix = np.random.choice(range(vocab_size), p=p.ravel())
    x = np.zeros((vocab_size, 1))
    x[ix] = 1
    ixes.append(ix)
  return ixes
```

The output we get trying different datas:

```
data has 633 characters, 49 unique.

alpha=0.001:
eFad-xCcSNQFbUvPQsnCx&tQLqcSgvfv&icJd.GvwV ’N
w;uKiLd’Wyj

FKvp&dwp;AwrZ.fw
xP;vwQbJTLS
s!
T
NsbeYAE ovEcfrIaYurpEZIpkJPt
B?QpVuNsmsk’Tn;:ZpyjzWysr G:aq.b:xHy
u&qsvMI
a
eO!vQsZYj-rrKifxq KBK?CuSSuUQ

alpha=0.1:

UEsxu.tw?JAo’cbmiG’
WqSBfp.msgNMw!WOo cyib’’r:
a!Y&LAMNBAYkBOT
dO?oi.tstajlonYmK-Kg’gBT,OpCD’
.?dosSeu.k?.Q&nh eoryfhPtWrwivih
Yedye&wpbl-oUVhgoxld Rn&RIhioqy.!qeSro.HiRNM?CgnfkpcQl!horrJZog; Aq.t s

```

```
alpha=0.3:
ee--dy fqCitsshiid bet'Br etve gwHry's gkd;n-basbqaia'snak: hafo.:
gets FiyOr ift:wy ax't,otee I
wk-pygre
Maln-noc?
Now;!-yIBe loecouy.l.t'dn''frire'' prkt'd.

Pdrm;' Thusi?buv:

OBUpd,
Obt.!,--w'st


alpha=0.6:
oldfreteatsro!
Ismm omabagng lovingsbep hatseat SgaRevon myol? gnme vigem, jusc brpe?s?' thou anome
thougiss me;-in! At yaD isfar as Aulsce:.

SICINIUS:
Hossizenvanbae'sri I he mied,
An muck we kine?

alpha=2:
ald 'tis love and the the he there have to be she of the me the should you so the che
to that the the to and in would let I the when, the site, the must he he my the be
the the this you have hows be

alpha=3:
an man the we the noble he when the the was the the he the the of he he he the son he
had the death come the gods the and and be of a mees hath he the condeth the we the
come and and the the the the

alpha=4:
an my and and the deak he and the the the the so stand and the conder the the we the
the so the the we and the the we shall the my the stand the the the the the where
the shall the shall the the the

alpha=5:
an my the the the the come the we the the the so have and a mee the the the come the
the the the the so the the the stand the we the the shall the the the when the son the
the the the the so the we t
```

When the alpha is too small the output does not make any sense at all, it's full of random characters put together. The best alpha is when it is set to 2, it has lots of words and the sentence is sort of making sense. When we set alpha bigger than 2, the are too many repeated words mostly "*the*".

Part 2

I copied a few lines of code from *lossFun*, code specified below.

```
def lossFun(inputs, targets, hprev):
.
.
.
.
  for t in xrange(len(inputs)):
    xs[t] = np.zeros((vocab_size,1)) # encode in 1-of-k representation
    xs[t][inputs[t]] = 1
    hs[t] = np.tanh(np.dot(Wxh, xs[t]) + np.dot(Whh, hs[t-1]) + bh) # hidden state
.
.
.
.
```

```
Interesting outputs:\\
1.\\
input: \\
And to make us no better thought of,\\
output: \\
you the the stare be my who that all the word of \\
Interpretation: It is a sentence kind of making sense.\\

2.\\
input:\\
his old hate unto you\\
output:\\
m!\\

CORIOLANUS:\\
I be the word, and he the cound th\\
Interpretation: It is a strong sentence so added a "!"\\

3.\\
input:\\
Hostilius\\
output:\\
 where and the strepere.\\
\\
MENENIUS:\\
I'll fars done\\
Interpretation: know it is a word, so added a space and finished the sentence.\\

4.\\
input:\\
COMINI\\
output:\\
US:\\
Now all lords math to the state thy the who be\\
Interpretation: Finished the name and detected it is a person.\\
5.\\
input:\\
Beauty\\
output:\\
: I can the of word with the to the all good the s\\
Interpretation: added ":" so know is a character's name and outputted a sentence in first person pers
```

Part 3

Probability of next char:

```
[[  9.99999994e-01]
 [  2.58678468e-23]
 [  6.38218866e-09]
 [  1.52872547e-22]
 [  1.07040707e-39]
 [  1.34919482e-20]
 [  5.23836513e-18]
 [  1.78045244e-16]
 [  7.17487588e-21]
 [  3.32908671e-15]
 [  8.88843909e-22]
 [  1.78563277e-22]
 [  1.68375774e-25]
 [  1.45807315e-31]
 [  3.45259119e-27]
 [  9.81251252e-42]
 [  5.97920874e-42]
 [  6.03386415e-29]
 [  3.40741694e-22]
 [  1.27435206e-36]
 [  7.28392823e-35]
 [  3.37225655e-38]
 [  4.30598823e-31]
 [  3.74103467e-34]
 [  5.10497190e-32]
 [  3.84603014e-30]
 [  7.37852309e-35]
 [  1.99676534e-32]
 [  6.85874822e-39]
 [  4.35673014e-40]
 [  1.24240074e-20]
 [  1.07844143e-35]
 [  2.28178946e-32]
 [  5.00473784e-31]
 [  8.06397489e-35]
 [  1.29698032e-39]
 [  9.53877444e-26]
 [  2.43848501e-31]
 [  7.89308440e-31]
 [  3.09387942e-27]
 [  3.49152957e-29]
 [  4.56777961e-30]
 [  8.80435152e-25]
 [  7.04104450e-23]
 [  4.89846718e-25]
 [  4.71900935e-26]
 [  3.22338147e-41]
 [  1.21973180e-23]
 [  4.52667541e-28]
 [  9.33898636e-31]
 [  3.50832293e-26]
 [  1.34161039e-38]
 [  2.34911298e-24]
 [  3.55929336e-18]
 [  3.88769300e-22]
 [  1.57980515e-27]
 [  9.83524409e-22]
 [  4.83881191e-28]
 [  3.87180522e-34]
 [  3.66808791e-29]
 [  4.95195866e-35]
 [  1.53710460e-27]]
```

It's obvious to see from all the probability listed above, the highest one is the char at index 0, and the second highest one is at index 2. Which is the new line character and the space character. Since I only set the column 9 of x to 1, after the dot product, only the column 9 of Wxh(Wxh[:,9]) will remain effective, which would matter in the result. And the row 0(Why[0,:]) and row 2(Why[2,:]) of Why determines the high probability of new line and spaces after column too.
The new line is $Wxh[100, 9] = 4.82918986837$, $Why[0, 100] = 2.67271236039$. The space is $Wxh[73, 9] = -5.05709665402$, $Why[2, 73] = -4.087729194$. Since they have the largest product and second largest product among all the other product for new line and space.

Part 4