Tobby Lie
CSCI 5931
Programming Assignment 3 - **Task 4**
10/27/19

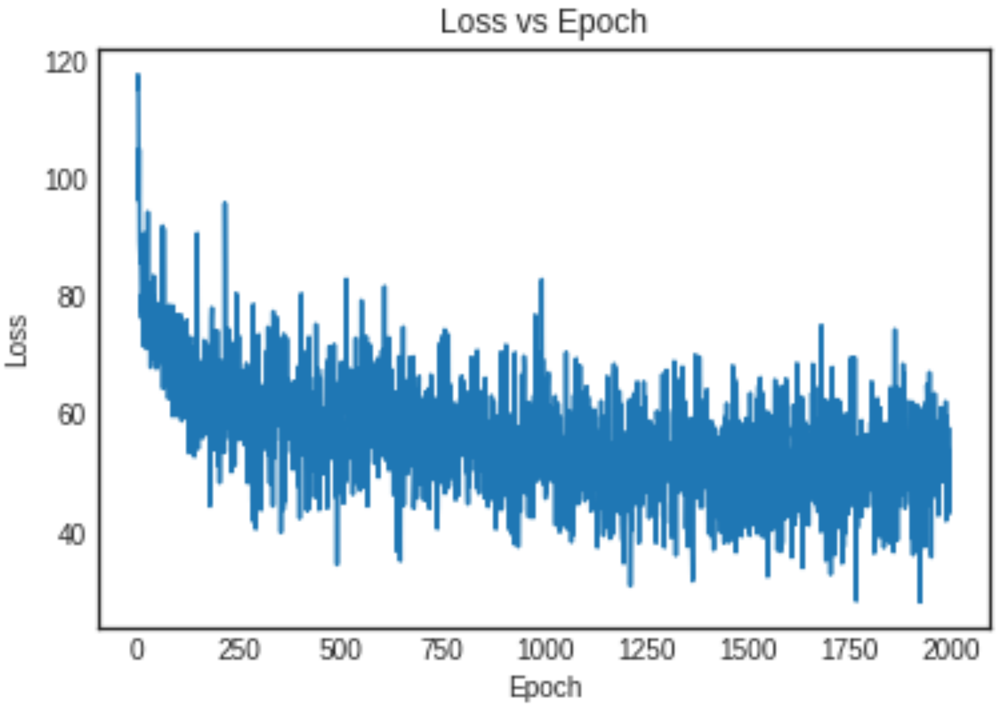

Figure 1: Loss vs. Epoch plot, training on 2000 epochs with hidden unit size 100

Generated sequences from training on 2000 epochs with hidden unit size 100 choosing 5 break points:

<u>400 epochs:</u>

```
 balloent. of vishave the gishers
atsen. the
ballopesed ssinu to the 2pered. the
murced revert of the markanie ffom those bild, berowers holath of his the veatring, at
deathe no hous. that he dishing i
```

<u>800 epochs:</u>

```
 ze ackimined he ucpospicy
taking 1825:
dapsigad cape batt at nor had in efeath lerk on plance of taken of a arel,
a cun’rcond, been lind!” avoce?


“nontand in allrisst of asy the beate? sbwhom pushoch
```

1200 epochs:

```
 ably iq0it hersert ar sspepsseded, and forrgert, not hexs of a milibuby. by nor his
ebald a blited
pellers, the farse, by the ucencest, hid
’stompt, “ix phrest
wind to mayre?.


the cassicalinictl hind
```

1600 epochs:

```
 asped the pile, not is sacen be logst,
passuse sisce. thimever, to
neb!”


hores a piloch guised to a deriss
-he
rropurion
tiwh of the eenc, port. o” the n ightrong.


“tow to wnopely, deparion contore.
```

2000 epochs:

```
 wellwe spews an saa afclees houll a couth. a shideing

wet, ha swide forermed sean
of wtur was and of,
the of inming
seat saterted..”

th
or grghercured the sied croand.


“f tron to the was heme hung,
```

Gradient Checking:

We approximate the numerical gradients by changing parameters and running the model. We checked if the approximated gradients were equal to the computed analytical gradients (by back-propagation). This was tested with parameters picked randomly for each weight matrix and bias vector. When implementing back-propagation it is useful to implement gradient checking as well in order to verify that the implementation is correct. The idea is that the derivative of a parameter is equal to the slope at the point which we can approximate by slightly changing the parameter and dividing by the change. When we compare the gradient calculated using back-propagation to the gradient estimated, if there is no large difference then we are in a good place. The approximation calculates the total loss for every parameter so it is very expensive which is why it is a good idea to perform it on a model with a smaller vocabulary.

The output from gradient checking:

Format: (param.name, grad_numerical, grad_analytical, rel_error)

```
W_f (5.279333e-07, 5.285817e-07) => 6.131629e-04
W_i (4.601155e-05, 4.601128e-05) => 2.887094e-06
W_o (1.468052e-05, 1.467997e-05) => 1.891436e-05
W_o (3.219824e-06, 3.219255e-06) => 8.846840e-05
W_v (5.898642e-05, 5.898623e-05) => 1.582185e-06
W_v (-7.884644e-05, -7.884747e-05) => 6.504861e-06
b_f (-1.358401e-03, -1.358398e-03) => 1.030336e-06
b_f (3.727401e-05, 3.727433e-05) => 4.385449e-06
b_o (-1.077325e-05, -1.077432e-05) => 4.975875e-05
```

All of the gradients calculated and the gradients estimated do not produce large differences so we are in a good place.

200 Hidden Units:

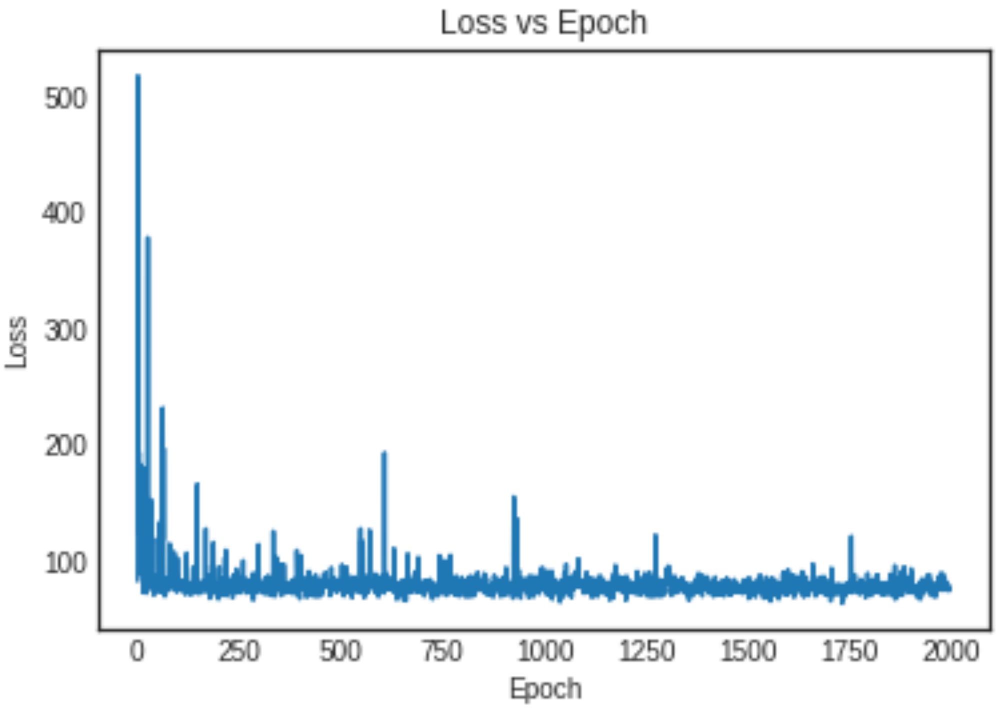

Figure 2: Loss vs. Epoch plot, trained on 2000 epochs with hidden units size 200

Text Sampling Results:

```
r, weriuteo sthaly th we th her mald and bleng, rethe dieed the semtreily somepey,
alned, on the scost which atr, his wan ;bsty themy ofed kipothesd, is a lot th al”
ofut “rent whicg fimed nkthey abd
```

50 Hidden Units:

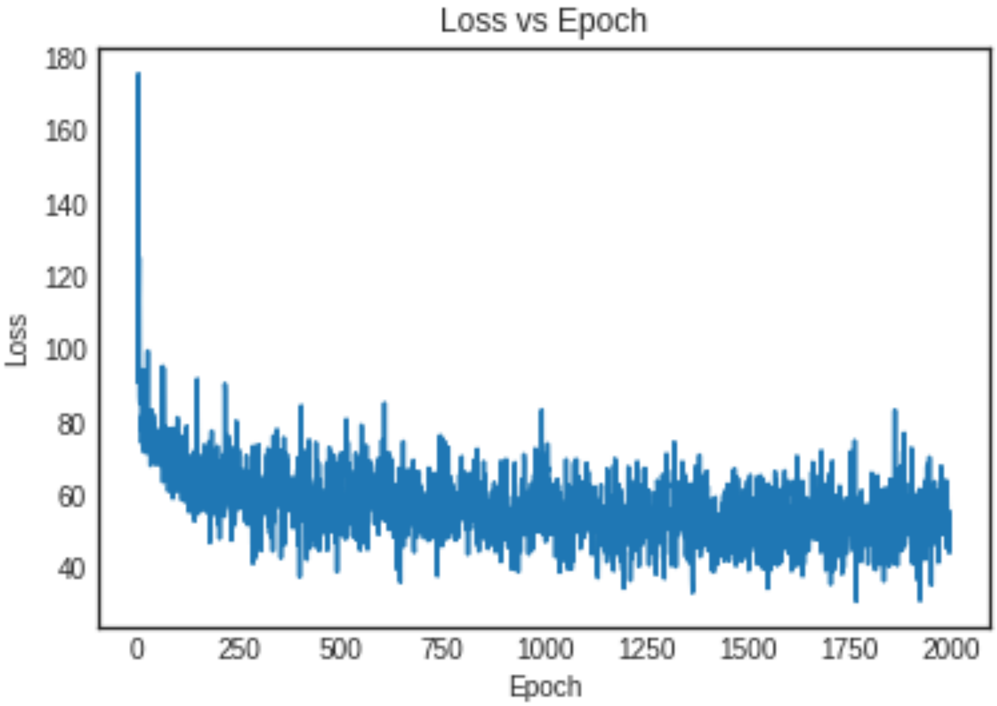

Figure 3: Loss vs. Epoch plot, trained on 2000 epochs with hidden units size 50

Text Sampling Results:

```
  dardend sed of op ortaet, aman
sould parmasathe the had sermiseily shimry sutd nissocs dimed the cliwe had a thee the
was an aj oflenrind thwlelr
mome erssomeo stly of it as itho lilled” he gondiof.
```

Discussion for part a:

In part a I experimented with doubling and halving the number of hidden units in our network. After training I plotted Loss vs Epoch graphs for each case and then generated text samplings from each case.

I observed that for a hidden dimension of 200, after 2000 epochs of training, the loss was reduced from approximately 500 down to 150 meaning a difference of 350. For a hidden dimension of 50, the loss was reduced from 170 down to 60 meaning a difference of 110. The hidden dimension of 200 yielded a greater reduction of loss meaning a smaller hidden dimension correlates to better reduction of loss for an LSTM. On the other hand, on a hidden dimension of 50, it was able to achieve a lower loss meaning more accurate results in the end.

In regards to the generated text samplings from each model it seemed that the model with hidden dimension of 50 produced more coherent text. Most notably it seemed that it was better than the model with hidden dimension 200 at producing longer sequences of actual words that also made sense slightly compared to the hidden dimension 200. The words in the hidden dimension of 200 were altogether more fragmented and disjointed than of size 50. Also, the sequences produced from hidden dimension of 50 were more complex in structure in terms of punctuation and newlines. A greater difference could probably be perceived with more epochs in training and with a greater difference in hidden dimension sizes.

Extended sequence length by doubling it:

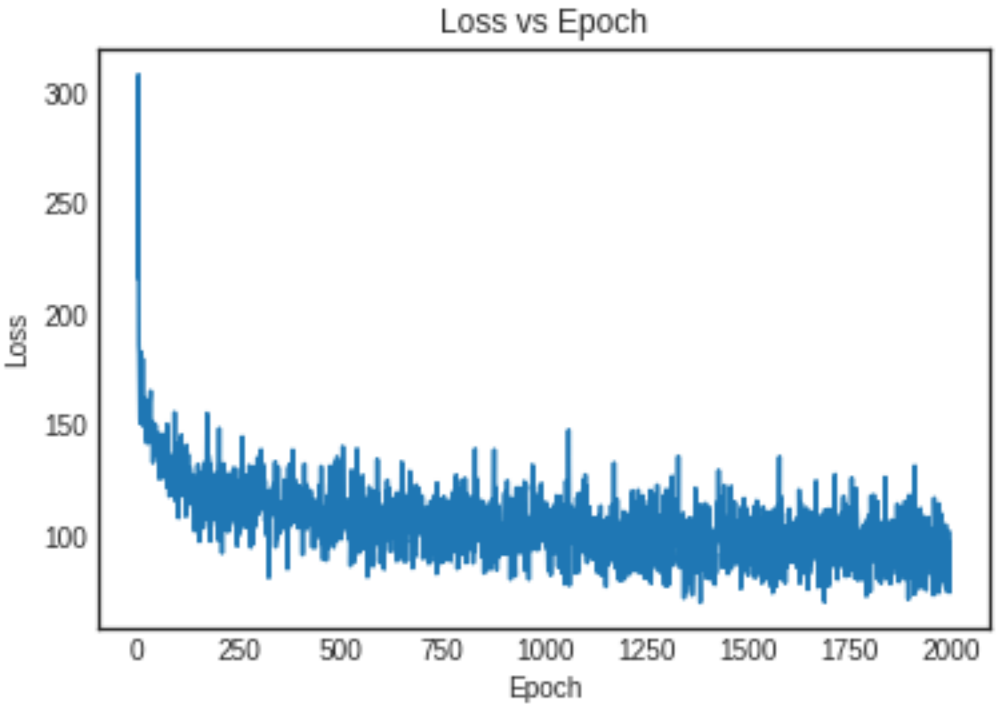

Figure 4: Loss vs. Epoch plot, trained on 2000 epochs with hidden units size 100 and double the length of sentence as original

Text Sampling Results:

```
 armilaler
the mysedd fero at ats, were rupparitsbess. mitadestone, with as to procgong is befn
not agsiryd thend reun lantert; as it what reade to know lavired who hipe magh of the
knobkt in. thac of
```

Shortened sequence length by halving each sentence to form a shorter sequence:

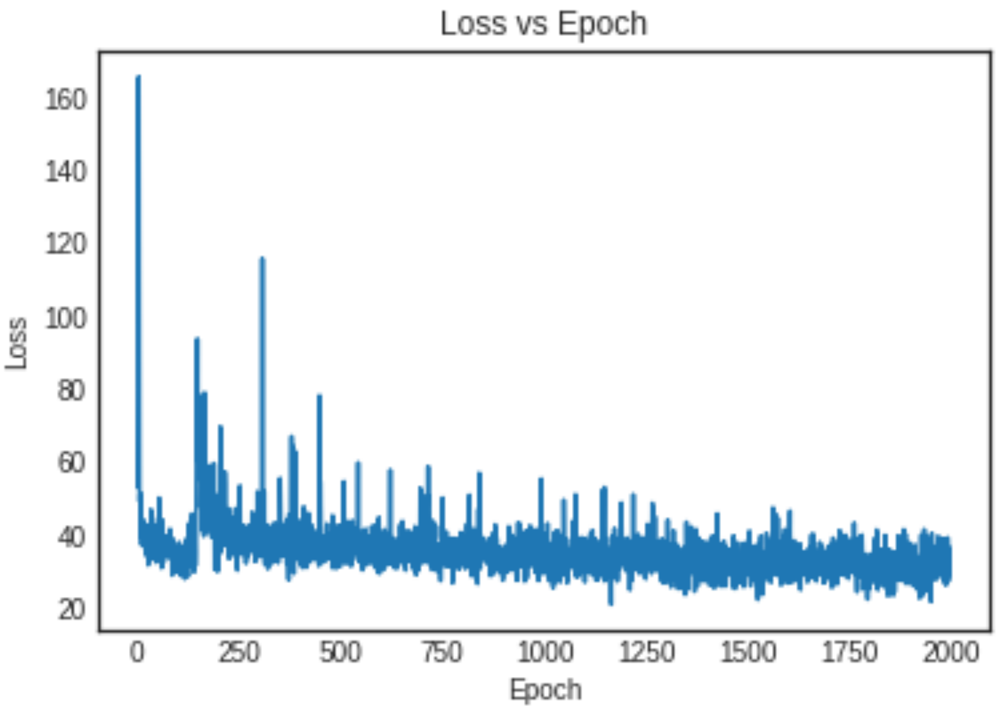

Figure 5: Loss vs. Epoch plot, trained on 2000 epochs with hidden units size 100 and half the length of sentence as original

Text Sampling Results:

```
   hif,l,y rig tdte lad ita ue dl thdo
routt vnygtot
d ifkd t bed he b ong aeofithenps, oi9onte tin besgboadoverd tiggrm mei b. bve winsis
 nw t agacer tertdttois anligiwdan in
d d g pfmbdt ve nta vig
```

Discussion for part b:

In part b I experimented with doubling and halving the sentence length being fed to our network. After training I plotted Loss vs Epoch graphs for each case and then generated text samplings from each case.

I observed that for the doubled sentence length training, after 2000 epochs of training, the loss was reduced from approximately 300 down to 100 meaning a difference of 200. For a halved sentence length, the loss was reduced from 160 down to 40 meaning a difference of 120. The doubled sentence length yielded a greater reduction of loss meaning a longer sentence length correlates to greater reduction of loss. This made sense to me as I believed a longer sentence length would mean that a network has more to learn from.

In regards to the generated text samplings from each model it seemed that the model with doubled sentence length produced more actual words rather than gibberish. In the halved sentence model, it seemed to produce less actual words but produced interesting sequence structures. A greater difference could probably be perceived with more epochs in training and with a greater difference in hidden dimension sizes. This outcome I also did expect to happen, similar to the previous paragraphs findings.