**Original networking with provided scripts:**

I combed through the cat\_in\_the\_hat.txt corpus just to double check for mistakes (and reread a classic at the same time), and I did find a typo on line 78, where

with a book in one hand!

was typed as

with a book one one hand!

Training the neural network took about 30-40 minutes, and produced the following results:

Seed:

" the ball!

but that is not all.

oh, no.

that is not all...'

that is what the cat said...

then he fel "

l on his head!

he came down with a bump

from up there on the ball.

and sally and i,

we saw all the things fall!

and our fish came down, too.

he fell into a pot!

he said, 'do i like this?

oh, no! i do not.

this is not a good game,'

said our fish as he lit.

'no, i do not like it!

not one little bit!

and then something went bump!

how that bump made us jump!

we looked!

then we saw him step in on the mat!

we looked!

and we saw him!

the cat in the hat

and he said to us,

'why do you sit there like that?'

'i know it is wet

and the sun is not sunny.

but we can have

lots of good fun, if you wish,

with a game that i call

up-up-up with a fish!'

'put me down!' said the fish.

'this is no fun at all!

put me down!' said the fish.

'this is no fun at all!

put me down!' said the fish.

'this is no fun at all!

put me down!' said the fish.

'this is no fun at all!

put me down!' said the fish.

'this is no fun at all!

put me down!' said the fish.

'this is no fun at all!

put me down!' said the fish.

'this i

Done.

Running the generator a few more times led to similar repetitive results, which makes sense considering the corpus itself is both repetitive and short. This more often than not led to some eerie “all work and no play makes Jack a dull boy”-esque results.

After running the generator for the first time, I read the warnings Python (specifically TensorFlow) was giving me at the beginning of each run and noticed I could use GPU acceleration for much better results. Upon installing everything I needed, I reran the default scripts and trained my neural network in a staggering 2 minutes.

**Change 1: Different corpus**

I was planning on including a short snippet of the script from the movie Ratatouille, since that is my partner’s all-time favorite movie, but considering the amount of optimizations using my GPU instead of my CPU achieved, I figured I may as well use the entire transcript (source = <https://www.scriptslug.com/assets/scripts/ratatouille-2007.pdf>).

Sanitizing the Ratatouille transcript took slightly more effort than I had originally anticipated, as I’m still not as fluent in Python as I am with Java, but Python is much more accessible for writing quick scripts, such as sanitizing inputs. Once I got everything working, I was able to easily remove all the page numbers from the text file using the following script:

if \_\_name\_\_ == '\_\_main\_\_':

with open("ratatouille\_script.txt") as original:

new = open("ratatouille\_script\_adjusted.txt", 'w')

nums = open("numbers.txt", 'w')

for line in original:

if line[0].isdecimal() and not line[1] == ')':

nums.write(line)

continue

new.write(line)

original.close()

new.close()

nums.close()

This allowed me to rewrite the transcript to a new file, as well as writing lines with page numbers to a separate file to verify I was not excluding any lines from the movie that happen to start with a number for some reason.

Lastly, I had to convert all the rich text characters “, ”, and ‘ to the Unicode version so Python could properly read them. Using the simple Replace All tool from WordPad did the trick, since Microsoft Word refused to use anything but the rich text version.

This corpus took about 35 seconds / Epoch for a total of roughly 30 minutes to train, a task that probably would have taken 9-10 hours without GPU acceleration.

This is the initial run after training using the entire transcript:

Seed:

"his toque.

linguini

and that's another thing. your

opinion isn't the only one that

matters here. co"

lette

wiat are you doing?

remy

(v.o.)

i was revrrnedt mine i mave and the ratt in the satt it the sat watis a bienc on the boy with a stueaking sound. she lady from the cookbook in a shnr of sat poated sight. dolette shene is the boy with a siall of the sound of she sat of the sat oo the boy with a stueak falls of the sound of rats ase gord and all to the soup. the rats to sake the satctouille with a start are gold from the satr to the sats to the coor into the satr to the sats to the coor into the satr to the sats to the coor into the satr to the sats to the coor into the satr to the sats to the coor into the satr to the sats to the coor into the satr to the sats to the coor into the satr to the sats to the coor into the satr to the sats to the coor into the satr to the sats to the coor into the satr to the sats to the coor into the satr to the sats to the coor into the satr to the sats to the coor into the satr to the sats to the coor into the satr to the sats to the coor into the sa

Done.

I problem that I never really noticed with The Cat in the Hat corpus is that oftentimes real words are not actually generated, which I recognize is a result of generating text one character at a time rather than using phonemes or even entire words, which would require gigabytes of training data and thousands of epochs to resemble real speech.

This generator also seems to get stuck loops towards the end, though I’m not sure what’s causing that, as it’s clearly not the corpus.

My python.exe crashed while I was training this network just as it got about halfway through, but I was able to load the most recent checkpoint and finish the remaining 24 epochs, which was truly a godsend.

**Change 2: Decrease the size of the sliding window**

This was originally going to be the first change I made to the network, seeing as the neural network would start simply restating large blurbs from The Cat in the Hat unchanged, but because of the aforementioned speed-up, I was much more excited to alter the corpus entirely to see what I could achieve with a larger vocabulary.

Having changed the seq\_length down to 20, the results became...interesting. This was the first gen I got:

Seed:

"should we tell her about it?

now, what should we do?

well...

what would you do

if your mother aske"

som things had to stop.

they whould wou do if your mother is oo her way to hav hite to the cat.

'i know it if sot of hoo out oot shipe to wlay will she say?

oh, the will not like it,

not ohem is oot all...

nook at me!

look at me!

look at me!

look at me

look at me

look at me

[...]

look at me

Done.

This was definitely the funniest text generated so far, and ominous to boot. I also realized that the seq\_length is defined in both the training script as well as the generative script, and upon having changed it in both places, I got this gen:

Seed:

"come here to play."

they will give you some fun

on this wet, wet, wet day.'

now, here is a game that they like,'

said the cat.

'i know some noo ohisgs

with thi dons

that are yit tack up th th.t.

thosl the way that they play!

if mother could see this,

oh, what would she say!'

then our fish said, 'lo! no!

make that cat go away!

tell that cat in the hat

you do not want to play.

he should not be here.

he should not be about.

he should not be here.

he should not be about.

he should not be here.

[...]

he should not be about.

he should not be here.

he sho

Done.

This one was interesting in the fact that the repetitive cycle was not a single line, but two, and it makes sense how something like this would start, considering the line from the corpus ‘he should not be here’ is followed by ‘he should not be about’, which is then followed by another ‘he should not be here’ due to the cadence and voice of Dr. Seuss. Reading through the corpus before training the neural network, I fully expected this to happen eventually.

I also like to think that ‘I know some noo ohisgs with thi dons that are yit tack up’ is something Dr. Seuss could have legitimately written.

**Change 3: Alter the dropout rate and number of epochs**

Originally, I was going to remove the second LSTM and Dropout layers, but I kept getting errors where

1) the shapes were incompatible

2) a number (typically in the 2000s) could not be converted to a char

3) layer mismatching (expected 0, saw 2)

4) unable to load weights into a subclassed model which has not been created yet

5) cannot reshape array to \_\_\_

I tried finding solutions to these errors, but I ultimately determined I could better spend my time altering different aspects of the model, so I settled on doubling the dropout rate and number of epochs. Here was the result:

Seed:

"e and thing two!'

so, as fast as i could,

i went after my net.

and i said, 'with my net

i can get t"

hem i bet.

i bet, with my net,

i can get those things yet!'

then i let down my net.

it came down with a plop!

and i had them! at last!

those two things had to stop.

then i said,to the cat,

'now you do as i say.

you pack up those things

and you take them away!'

'oh dear!' said the cat.

'i will not let you fall.

i will hold you up high

as i stand on a ball.

with a book in one hand!

and a cup on my hat!

but that is not all i can do!

said the cat...

'look at me!

look at me now!' said the cat.

'with a cup and a cake

on the top of my hat!

i can hold up two books!

i can hold up the fish!

and a little toy ship!

and some milk on a dish!

and look!

i can hop up and down on the ball!

but that is not all.

oh, no.

that is not all...

'look at me!

look at me!

look at me now!

it is fun to have fun

but you have to know how.

i can hold up the cup

and the milk and the cake!

i can hold up these books!

and the fish on a rake!

i can hold the toy ship

and a little toy man!

and look! with my tail

i can hol

Done.

This seemed to be the most comprehensible model and didn’t end with a cyclic stream of terrifying nonsense. Obviously this time around, the neural network is spitting out large chunks of unchanged corpus, though I was surprised when it suddenly switched to a completely different part of the text rather than what normally comes next, so it’s not entirely overfit, likely due to the 0.4 droprate, but I bet the neural network would perform better (more original) with fewer epochs.