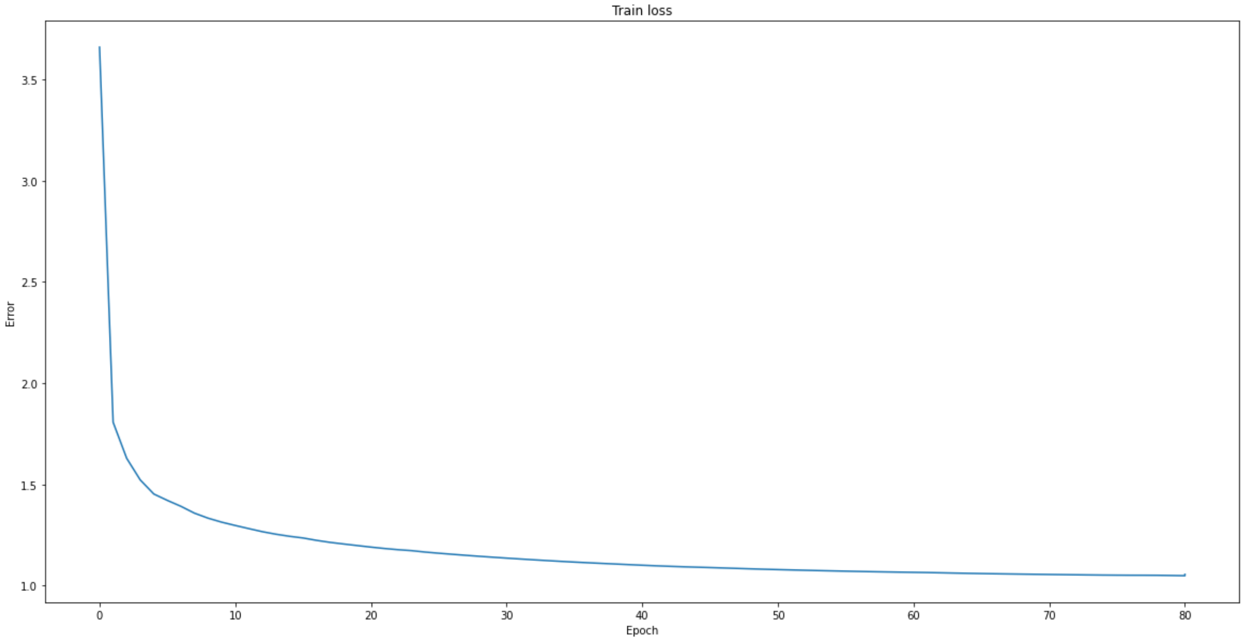
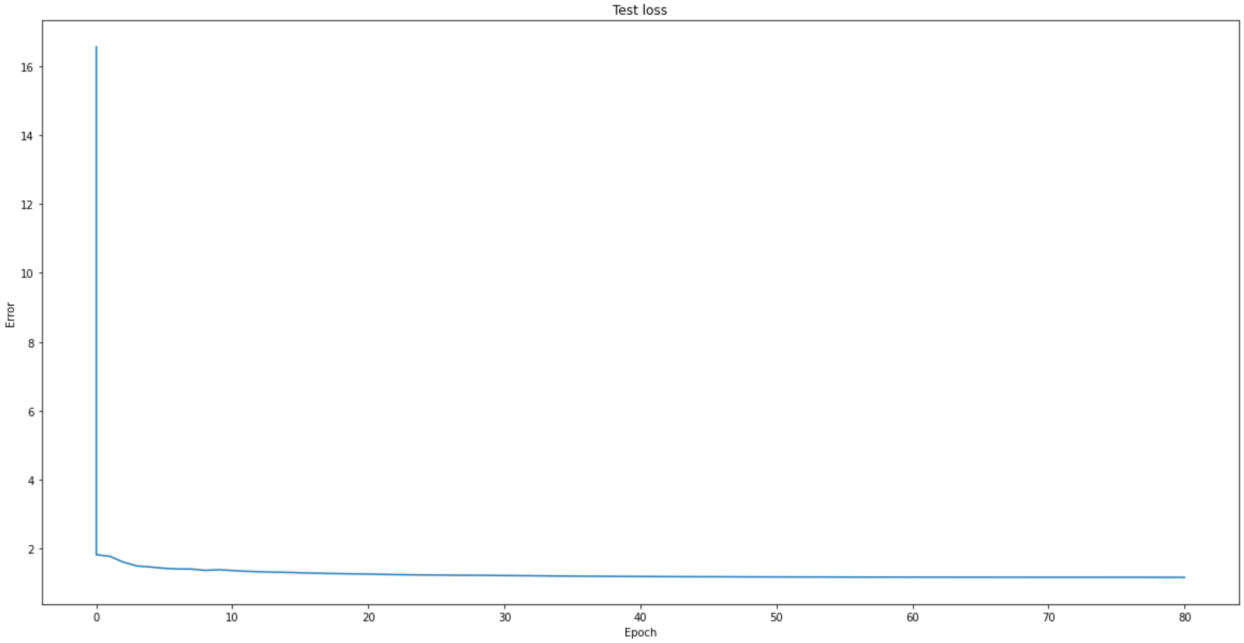
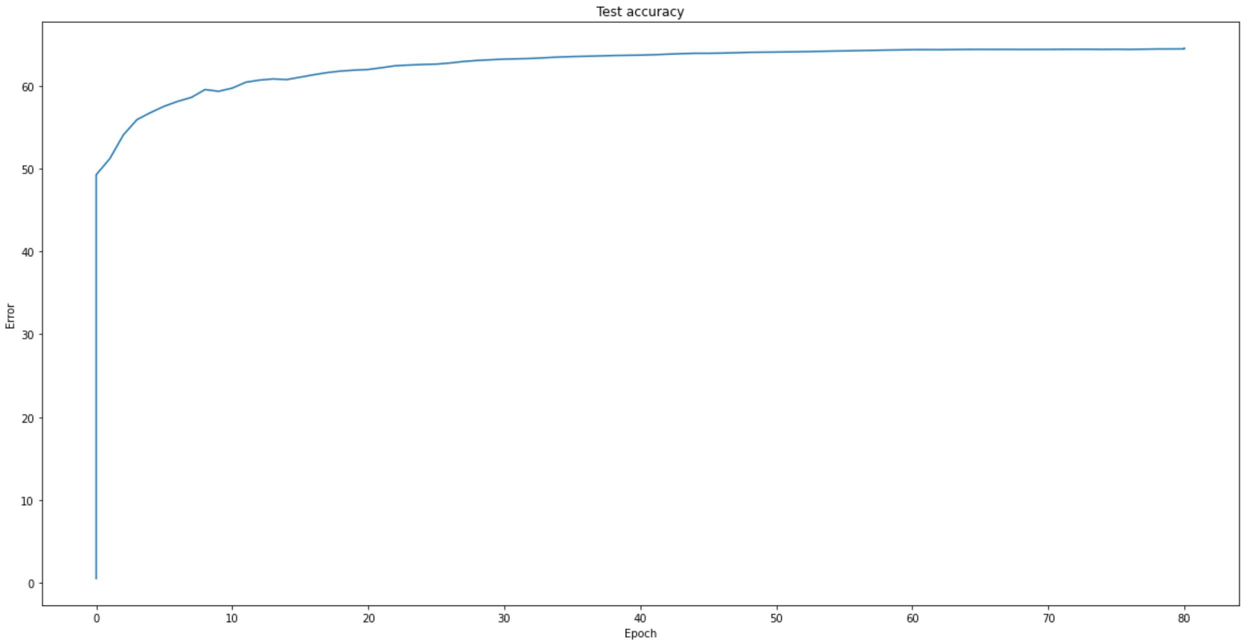
CSE599G1 (CSE490G) Deep Learning - 20AU
Will Chen, Joyce Zhou

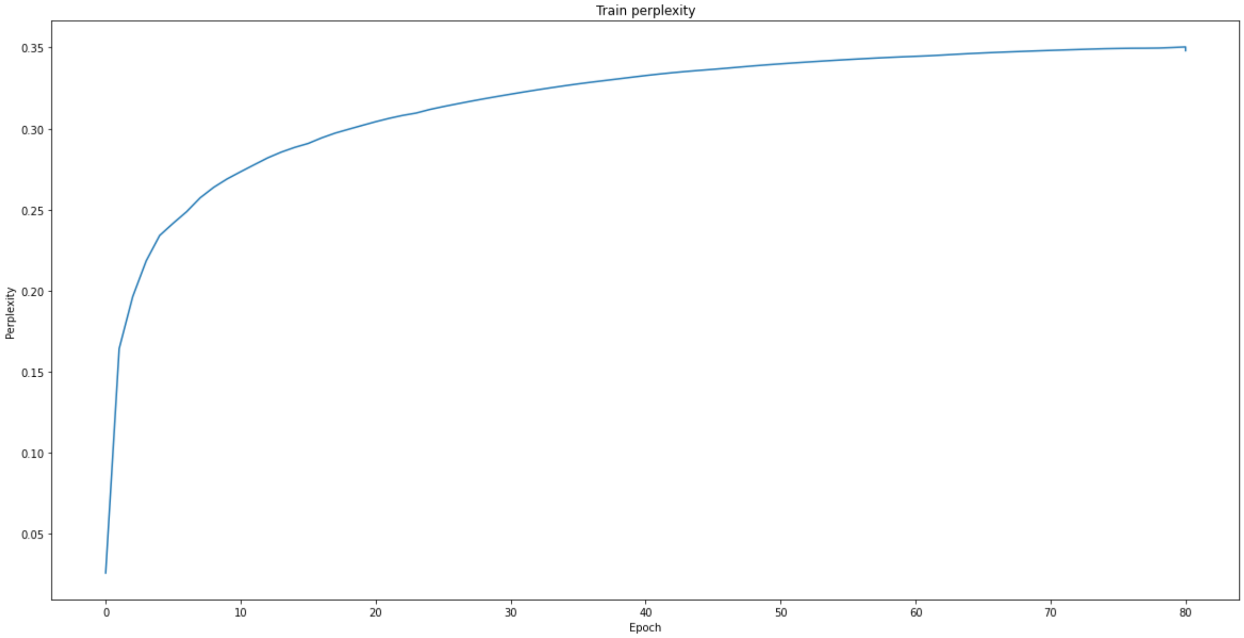
HW2 Short Answer Questions

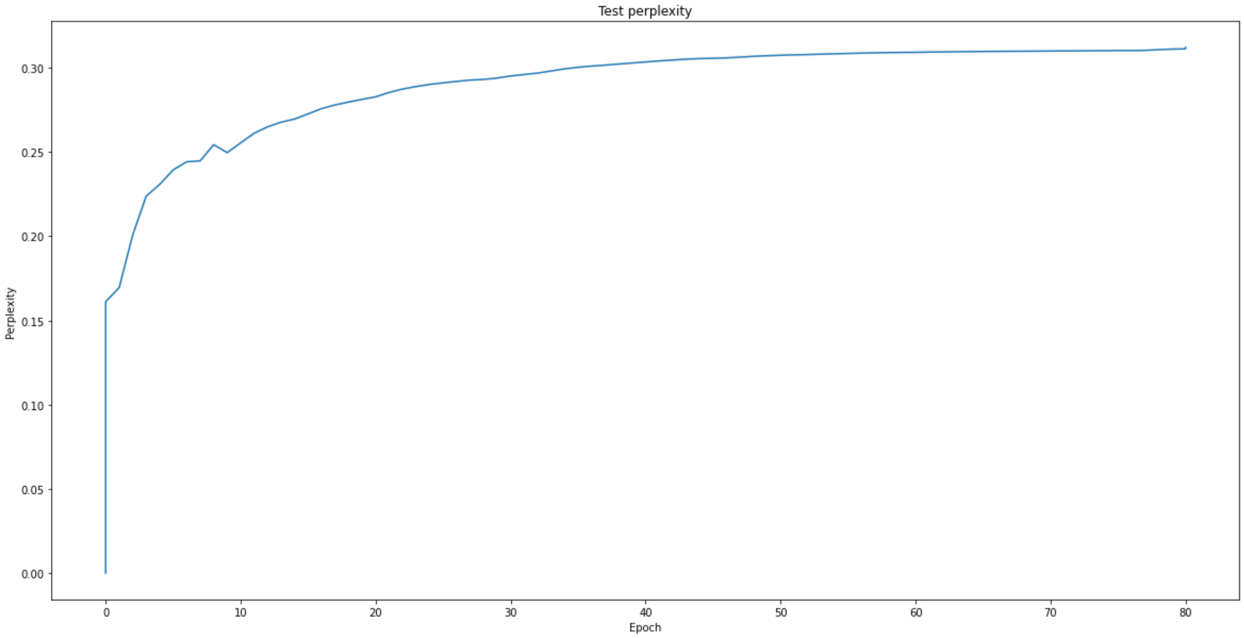
1. **Plots for training error, test error, and test accuracy. Also plots for train and test perplexity per epoch.**
    a. The best-performing model that we found had batch size 256, feature size 512, learning rate 0.002 (2e-3), weight decay 0.00005 (5e-5), and was trained for 80 epochs.






Test accuracy


Train perplexity



2. **What was your final test accuracy? What was your final test perplexity?**
    a. The final test accuracy was 64.48% (808873/1254400 test cases)
    b. The final test perplexity was 0.3112
3. **What was your favorite sentence generated via each of the sampling methods? What was the prompt you gave to generate that sentence?**
    a. Max (input “`"I am Voldemort."`”, temperature 1, sequence length 200): `"I am Voldemort." "I don't know what you're going to be able to see the party with your brother stuff," said Harry, staring at her. "What are you going to be able to see the party?" "I don't know what you're going to b`
    b. Sampling (input “`Voldemort told me`”, temperature 0.7, sequence length 200): `Voldemort told me what you're doing in the castle." "Oh, it's all right, Harry, I'd be Albus Dumbledore and he would be placed into the fire, they were all right, Harry, which was he looking up at the end of the desk,`
    c. Beam search (input “`"Hello."`”, temperature 2, sequence length 200): `"Hello." Harry stared at him. "What are you going to do with the other two worst things?" said Harry, who was looking at him with a large chair and saw that the wind was still smiling and slightly to his feet.`
4. **Which sampling method seemed to generate the best results? Why do you think that is?**
    a. Beam search produced the best results overall. Beam search and max selection produced the same results most of the time, but beam search allows for some randomness in generation while max selection is strictly the same for each model and input. This makes sense because beam search is designed to combine the best parts of sampling and max selection: it emphasizes selecting the most likely

output, yet can still explore the output space to catch sentences that max selection will miss.

b. (And there's still some random chance in beam search sampling, so it's more fun for us to read the outputs.)

5. **For sampling and beam search, try multiple temperatures between 0 and 2.**
    a. **Which produces the best outputs? Best as in made the most sense, your favorite, or funniest, doesn't really matter how you decide.**
        i. Joyce thinks that a temperature of 1.1 combined with the sampling method produces results that are sometimes obviously wrong but in ways that are pretty funny (generating words that don't exist but *could* exist, or are very easy to imagine reading in an accent). For results that are actually more readable, going with 0.7 is boringly reasonable.
    b. **What does a temperature of 0 do? What does a temperature of 0<temp<1 do? What does a temperature of 1 do? What does a temperature of above 1 do? What would a negative temperature do (assuming the code allowed for negative temperature)?**
        i. What we observed is that the lower the (positive) temperature, the less variation there is between different texts generated. This is most visible in the sampling generation method, where a low temperature results in almost all of the output texts being identical, while high temperature results in almost chaotic output. The reason why this happens is that in the inference function, the output vector is divided by temperature before applying softmax, so essentially the greater the temperature the more evened out probabilities become. A negative temperature would invert this, dramatically increasing the probabilities of characters that were originally "not predicted". Greater magnitude of temperature would decrease this effect as the softmax output makes it more "evened out" again.
        ii. A temperature of 0 actually acts identical to a temperature of 1e-20 in this code, since there's a hard limit on what temperature will be accepted in the model definition. However, theoretically, this would either dramatically emphasize the single most likely or the single least likely output character depending on how we approximate (from negative or from positive direction).
        iii. A temperature between 0 and 1 tends to increase the probability of characters that were already likely to be next / predicted and decrease probability of unlikely characters. It makes the output distribution peaks and valleys more dramatic!
        iv. A temperature of 1 outputs exactly what the model is trained to do without modifying any of the output distribution.
        v. A temperature above 1 tends to decrease probability of predicting likely characters and increases probability of unlikely characters. It evens out the output distribution so that sampling methods end up picking characters more randomly.

Hyperparams experimented with: (Harry Potter model, default architecture)

Batch Size	Feature Size	Epochs	Learning Rate	Weight Decay	Test Avg Loss	Test Accuracy (# correct out of 1254400)
256	512	20	2e-3	5e-4	1.3409	749730
128	512	20	2e-3	5e-4	1.3494	743984
512	512	20	2e-3	5e-4	1.3717	739476
256	256	20	2e-3	5e-4	1.3842	735663
256	1024	20	2e-3	5e-4	1.4797	703803
256	512	10	2e-3	5e-4	1.3809	735208
256	512	30	2e-3	5e-4	1.3242	753521
256	512	20	2e-4	5e-4	1.3654	738189
256	512	20	2e-2	5e-4	1.5105	685069
256	512	20	2e-3	5e-5	1.2596	776556
256	512	20	2e-3	5e-3	2.1634	488036
256	512	30	2e-3	5e-5	1.2107	793982
256	512	40	2e-3	5e-5	1.1932	798890
256	512	50	2e-3	5e-5	1.1804	803113
256	512	30	2e-3	5e-6	1.2732	779622
256	512	30	2e-3	5e-7	1.2820	782481
256	512	50	2e-3	5e-5	1.1838	801429
256	512	60	2e-3	5e-5	1.1762	803093
256	512	70	2e-3	5e-5	1.1684	806662
256	**512**	**80**	**2e-3**	**5e-5**	**1.1643**	**808873**
256	512	100	2e-3	5e-5	1.1715	805242

Other experiments

1. (New Corpus)
    - a. **What corpus did you choose? How many characters were in it?**
        - i. We chose to use the text of Worm (a *really* long web serial about superheroes written by Wildbow: https://parahumans.wordpress.com/) as an alternative text corpus. This included text from the author's notes and was formatted similarly to Markdown. The uncleaned corpus contains 9556529 characters. This decreases to 9399225 characters after cleaning up whitespace.
    - b. **What differences did you notice between the sentences generated with the new/vs old corpus?**
        - i. The new corpus (Worm) produces significantly more quotes, which is probably because Worm itself focuses more on dialogue than Harry Potter. The new corpus also leads to sentences that are a little more coherent, even at the same temperature. This is because the new corpus is larger (about 1.5x times the length) and also possibly because sentences in the new corpus are simpler to start with, meaning it’s less likely to wander. There’s also a lot more recognizable names and Markdown formats, likely because Worm refers to its character/superhero names extremely frequently and also makes liberal use of italics and other formatting (which got included in the txt file download).
    - c. **Provide outputs for each sampling method on the new corpus (you can pick one temperature, but say what it was).**
        - i. I used sequence_length=200, temperature=1.1 for these outputs. The input phrase for all of these was “`I cracked`”
        - ii. Max: `I cracked a little as the same thing about the same page that I was surprised to see the same thing. I was surprised to see the same thing that I was surprised to see the same thing that I was surprised to see`
        - iii. Sampling: `I cracked, another proper dumble. Other half-walk, cutcons and drawers or phins altered by the bitiness. It must enough and other than the blatants are much her topd. "Thouse thos: Simurgh for you going to fin`
        - iv. Sampling: `I cracked down at the back of it. The minute affected the buttness, simugpy to fight one of this. It was *Wards* of will, Mark and them was. Then I adjusted the thread one morbird tanoe comstruction to the gro`
        - v. Sampling: `I cracked kickiently, strapping the *sudden instead, keeping my key cheses up into the knife to the Blenden line fight on. So many. As you're`

thinking that the swattings increst it not in discream with it," Al

vi. Beam search: I cracked a little as the same thing about the fact that I was surprised to be a bit of a second to get a better power. I was sure what I was doing. I was sure what I was doing. I was sure what I was doing. I

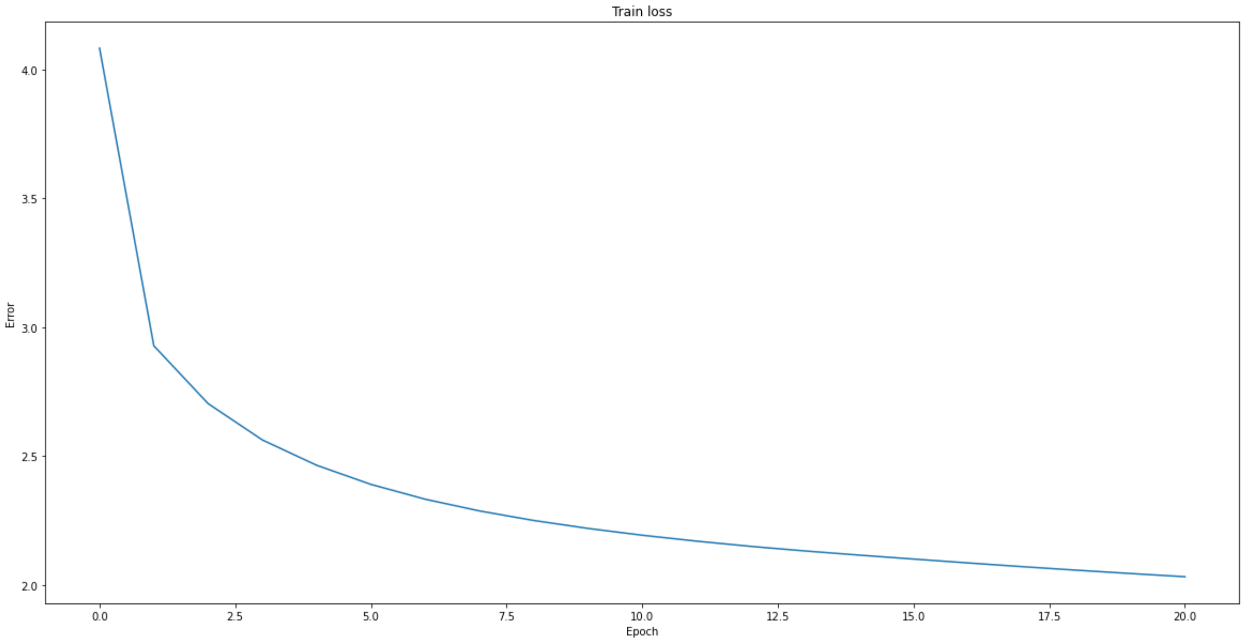
vii. Beam search: I cracked a little as the back of the building was almost a barrier against the street. I could see the strength to the back of the building and the bugs that were still contributing the streets. I could see t

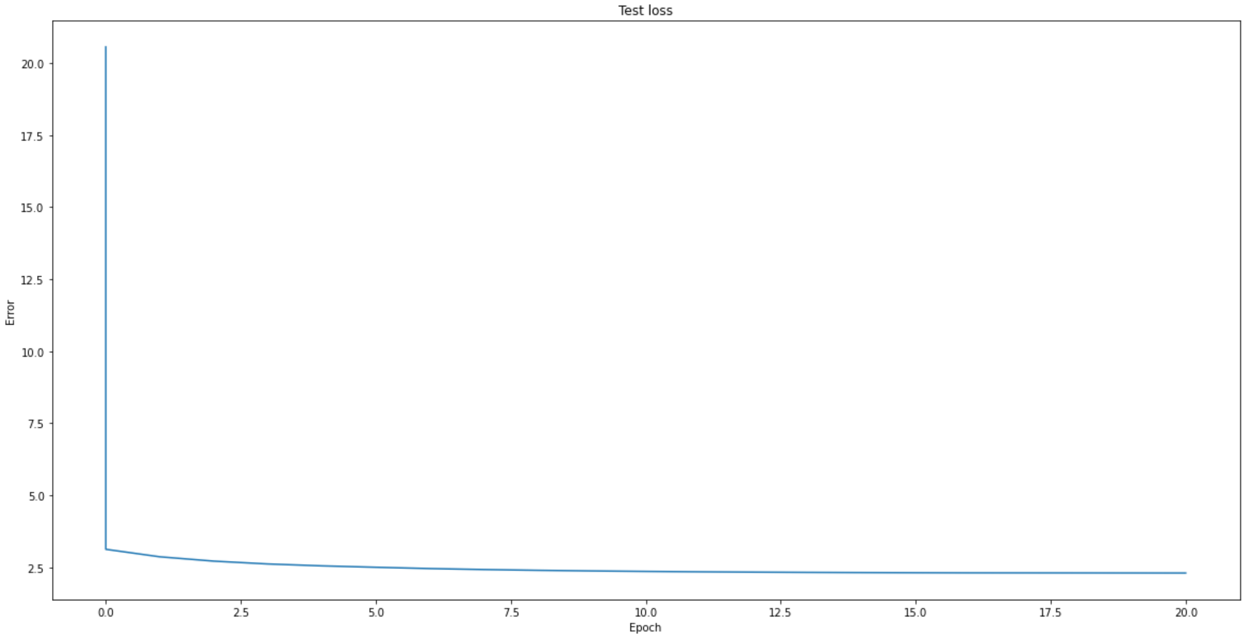
Batch size	Seq length	Feature size	Learning Rate	Weight Decay	Avg Test Loss	Test Accuracy (# correct out of 1894400)
256	512	20	2e-3	5e-4	1.3532	1105988
256	512	30	2e-3	5e-5	1.1855	1187999
256	**512**	**50**	**2e-3**	**5e-5**	**1.1607**	**1203047**

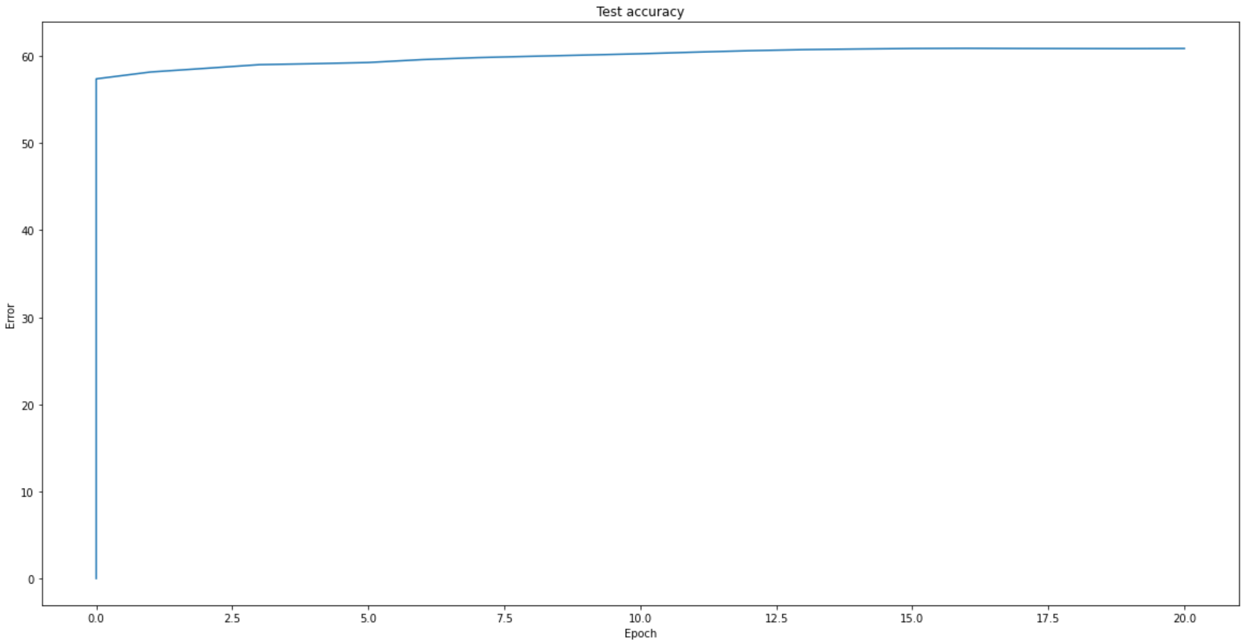
6. (Words)
   a. **What new difficulties did you run into while training?**
      i. Parsing the corpus into words posed some initial challenges, for example splitting by whitespace caused many grammatical and punctuation errors in the generated sequence. These errors arose because punctuation such as quotation marks at the beginning of a spoken sequence gets added in as part of the word that immediately follows it. We were able to solve this problem by using a simple regular expression to separate prominent punctuation symbols (such as ,.!:;?"-) with whitespace, then compressing and splitting on whitespace as we did before. Another difficulty was that the word-based model would cause CUDA to run out of memory, which we circumvented by decreasing the batch size from 256 to 128 (and smaller).
   b. **How large was your vocabulary?**
      i. Our vocabulary size was 29,432 words, which is a marked decrease from the 9,399,225 characters we had before for the character-based model.
   c. **Did you find that different batch size, sequence length, and feature size and other hyperparameters were needed? If so, what worked best for you?**
      i. Yes, as noted above, we needed to decrease the batch size in order to avoid CUDA running out of memory. We also tested a variety of other

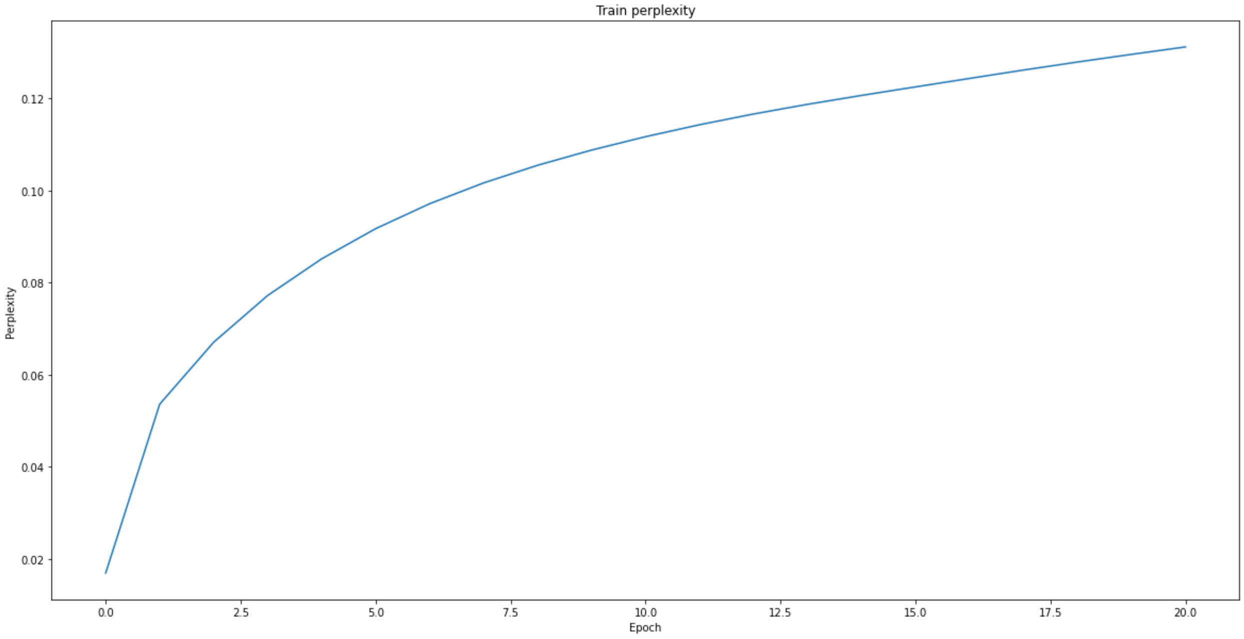
hyperparameter combinations, which are shown in the grid below. For the experiments shown below, we fixed the number of epochs at 20 because we observed that this setting led to comparable performance when compared to higher settings, and that other hyperparameters had a larger impact on model performance. We found that the best model used a batch_size=32, sequence_length=100, feature_size=256, learning_rate=0.001, and weight_decay=0.00005. Specifically, this model achieved an average test loss of 2.3112 and a test accuracy of 1012742 correct out of 1164000, or 87%. Plots for the best model are included below.

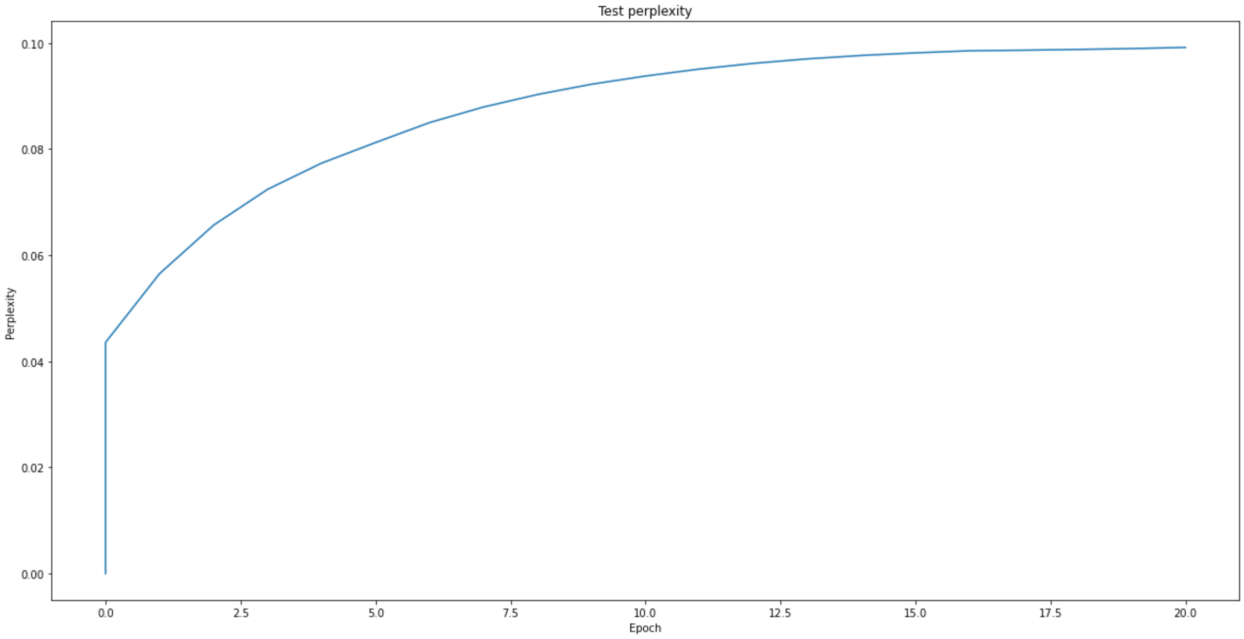
Batch size	Seq length	Feature size	Learning Rate	Weight Decay	Avg Test Loss	Test Accuracy (# correct out of 1664000)
32	100	512	0.002	0.0005	2.5394	993333
64	100	512	0.002	0.0005	2.5447	992249
128	100	512	0.002	0.0005	2.5833	985658
32	125	512	0.002	0.0005	2.5424	992309
32	150	512	0.002	0.0005	2.5437	991780
32	75	512	0.002	0.0005	2.5541	990839
32	100	256	0.002	0.0005	2.5318	995704
32	100	1024	0.002	0.0005	2.5815	985963
32	100	256	0.001	0.0005	2.5057	997726
32	100	256	0.005	0.0005	2.6058	989706
32	100	256	0.001	0.0001	2.3341	1010868
32	100	256	0.001	0.00025	2.4010	1006720
32	**100**	**256**	**0.001**	**0.00005**	**2.3112**	**1012742**
32	100	256	0.001	0	2.5808	998189


Train loss
Error
Epoch
4.0
3.5
3.0
2.5
2.0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0


Test loss
Error
Epoch
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0


Test accuracy
Error
60
50
40
30
20
10
0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Epoch


Train perplexity
Perplexity
0.12
0.10
0.08
0.06
0.04
0.02
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Epoch



8. (Sentences)
   a. **What new difficulties did you run into while training? What new difficulties did you run into while preprocessing?**
      i. The first major difficulty while preprocessing was related to parsing sentence boundaries. We used spaCy to parse the entire document to find sentence boundaries, however, its memory limitations meant that the document needed to be separated into multiple parts in order to be processed. We overcame this by adding an additional step in preprocessing to split the document on "\n\n\n" - these matching up with the boundaries between separate books in the Harry Potter series, and therefore guaranteed to match up with accurate sentence boundaries. We also raised the text length limit on spaCy to be able to parse entire books at a time. (Other split tokens weren't as reliable - we tried "\nCHAPTER", but OCR on books failed often enough that this still produced text over the length limit.) We didn't run into any new difficulties during model training.
   b. **Were the results better than with the original data loader?**
      i. We didn't compare performance of models trained with a sentence-recognizing data loader vs. the original data loader one-for-one on hyperparameters, but based on a few experiments, overall the performance of models trained with <EOS> characters did slightly worse than the original models.
   c. **Provide some outputs for each sampling method (you can pick one temperature, but say what it was).**
      i. I used temperature=1 for these outputs. The input phrase for all of these was "`Harry Potter and the`"

ii. Max: Harry Potter and the only one who was standing in the corridor that had been a gaze of parchment and a strange struggle of the silver stars and silent tables and the fire was standing in the door and started to see the b (it didn't reach an <eos> before 200 characters)

iii. Sampling: Harry Potter and the Snape.

iv. Sampling: Harry Potter and they were…

v. Sampling: Harry Potter and the owl is still looking ouncing humans.

vi. Beam search: Harry Potter and the only one who was standing in the corridor that had been a gaze of parchment and a strange struggle of the silver stars and silent tables and the fire was standing in the door and started to see the b (this is the exact same output as from max method - it didn't reach an <eos> before 200 characters)

vii. Beam search: Harry Potter and there was a silence fell on the floor and the only sound of the castle was still an old water.

Batch size	Seq length	Feature size	Learning Rate	Weight Decay	Avg Test Loss	Test Accuracy (# correct out of 1254400)
256	512	20	2e-3	5e-4	1.3682	739732
256	512	30	2e-3	5e-5	1.2463	782751
256	512	50	2e-3	5e-5	1.1931	800386