Classical Literature Generation Based on Character-Level Recurrent Neural Network

Buyan (Jacky) Li 205126244
William Scott-Curtis 905354911
Rehan Chinoy 405138632
Rebecca Guo 205208065

Ⅰ. Introduction

Our interest in text generation by recurrent neural networks began after reading a blog post by Andrej Karpathy [1], along with a companion tutorial in TensorFlow [2] which focused on simulating Shakespearean literature with a character-based RNN model. We decided to expand this model to work with two other authors: Homer and James Joyce. Both Homer and Joyce, like Shakespeare, have distinct styles, which makes assessing the quality of the simulated text as simple as possible.

We found that the model was able to produce text simulating the sample data, the epic poems *The Iliad* and *The Odyssey* by Homer, and the novels *Ulysses* and *Dubliners* by Joyce, reasonably well, and the distinct style of each author was broadly evident, and most of the words in the generated text were real words. However, there were few coherent clauses, and the sentences made no grammatical or practical sense. The accuracy of the simulation is hard to determine beyond these broad statements because we were unable to contact a domain expert (in this case, a literature expert specializing in Homer and/or Joyce) to verify the nuances of the style. However, the style of the generated text is quite similar to the respective authors' styles from a surface reading.

We further analyzed the RNN by changing the architecture of the hidden layer (we used both a GRU and an LSTM), the number of epochs, the length of the training sequences, the batch sizes, and the activation function. We found that out of these five conditions, only increasing the number of epochs led to a lower minimum training loss and qualitatively more accurate simulation of the text.

Ⅱ. Methods

Recurrent Neural Networks (RNNs) have been dominating the field of deep learning for sequential data due to their ability to maintain an internal memory and make predictions based not only on its most recent input but also its previous inputs.

We used the Gutenberg .txt files for each piece, trimming any text that wasn't directly from the poem or novel. Processing the data comprised of vectorizing text into character IDs and inverting those IDs into readable text using the Tensorflow string lookup layer.

We then broke the texts up into sequences of 101 characters each, where a training sequence includes the first 100 characters, and the target sequence includes the last 100 characters, with the goal of the model to predict the next character after the training sequence. We created batches of 64 of these sequences to train the model on.

The model consists of three layers: an input layer that maps each character to a vector, an RNN (we use a GRU and an LSTM, or long short-term memory), and an output layer, a vector containing the log-likelihood that the next character is each character in the vocabulary. By implementing the GRU or LSTM layer, we are able to avoid the vanishing gradient problem by being able to better retain information long-term via LSTM's forget gate and GRU's update gate due to an additive rather than nonlinear update function [6] [8]. We treat the model as a categorization model to train it, using the sparse categorical cross entropy loss function and the Adam algorithm to optimize. To begin, we trained on each dataset for 20 epochs, and used the train models to generate the attached text.

In order to further understand how the parameters of the model affect the simulation, we selected *Ulysses* and changed the activation function used in the model, the type of RNN, the size of the training sequences and the batches, and the number of epochs during training, and recorded the changes in the loss, and generated more text to see any changes in quality of the simulation.

Ⅲ. Results

The predictions are generated by taking a character input as the very first letter of the predicted text. Here are the generated text after training on each of the classic works:

Ulysses:
Duke mammad
no hurry to throw for, throb in his mina border. What do they invent out
too downs what means to keep them to defull to think of home shewness.

—Who is he knew, Stephen interfected his armpit, Greem house. Two and
pinstrite of it. It was dear gord.

Dubliners:
Duke pulling knee. To express through the respectfull of
Sinceralt from Bridge they used to puff your took let his nose in by
downstairs upoticular whose latesting for their little old father involdst by her
hat into easisty. Little Browne, both began to money on to it in to

The Iliad:
My
friends with many a wall as one that has been many flying pale within reach, while Patroclus
to Idomeneus having
given overtake me back to Ilius, but my

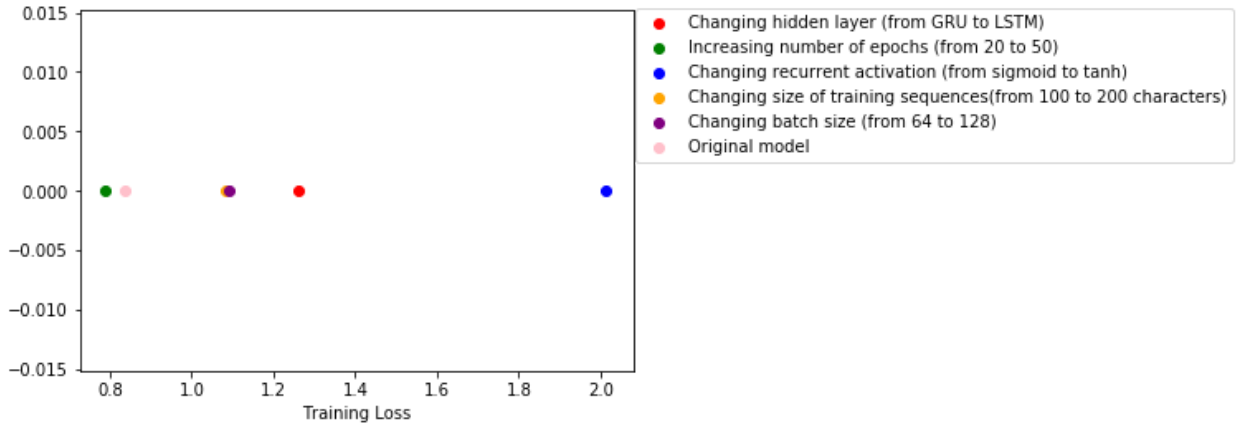
side was
he that had fallen

The Odyssey:
Telemachus, "Stranger," said Ulysses,
"dy quiver there is nothing better of arrows. Everything as she took his head and bronze
with his old day."

As we can see, most words in the generated text are indeed actual words in English, and some of the phrases are coherent. However, none of the sentences are grammatically correct nor make any sense in terms of its meaning. This means that our character-level RNN, by learning the probability of the appearance of a certain character based on the characters before it, was able to capture the structure of the words but not the sentences. Naturally, this will be our next step of improvement. As of right now, however, we still haven't seen a deep learning model that can effectively capture both the word and sentence structures.

While we had difficulty evaluating the outputs of our model in a rigorous and reliable manner, we wanted to adjust certain parameters to see if we could capture any difference in performance. These parameters included the hidden layer we were using, the number of epochs the model was trained for, the training sequence lengths and batch sizes, and the activation function used. Measuring a change in training loss between our original model and our adjusted models seemed to be a plausible and empirical test of how our parameter changes were performing against the training data. Below, we have included a table of our results after changing those parameters for our Ulysses text, which we assume will behave approximately the same with our other three bodies of work. However, it should be noted that measuring training loss alone does not provide a rounded image of how well the model produces new text, as well as ignores potential concerns of overfitting. As a result, we would also consider validation loss and other indicators of general performance as a future step.

Parameter Change	Training Loss Before	Training Loss After
Changing hidden layer (from GRU to LSTM)	0.8362	1.2594
Increasing number of epochs (from 20 to 50)	0.8362	0.7897
Changing recurrent activation (from sigmoid to tanh)	0.8362	2.0146
Changing size of training sequences (from 100 to 200 characters)	0.8362	1.0841
Changing batch size (from 64 to 128)	0.8362	1.0901



Ulysses (50 epochs):

Duke strikes over the
ship and the diaphane in Mr Bloom's embty police doubled in her
eating pierce amid with sports of course with which are
their registed spareller,
muchled abart the leaves off. Recent at the child was high it in
me. In. You bring the place but Hannaphaid is tough nouse to convent. Padding
in the bed. Penders for some mortar he kissed the sheet is it. That was
chargebour. Four poetry figures for the tramline if all the

Ulysses (tanh recurrent activation function):

Duked puld _nester Malw and to know of eve poly
seellw with time in and turned Marond

shan this eyes and fecernugulut
to muss Conty the _amérsemede iman's awmy Madend mid
mateswer dlind, madtubfer cofoner.

—Greet
blems the litsteraw a doush, up the sming eagoret.
M... Wast eve milite.)_ Top to the mold whiece and the unfurd of crarly seatron and take whinived of of the

Ulysses (LSTM):

Duke f I h Hevee inimoflingofuss ushe cheshe f cse Du
Blyom BLoflay the atrnths alle, hed were, pe. n w Sthensheng r he rs By ngrk I ody t ok thean m,, of estate.
melug ang ile lly Loun verensis than ir Tofrinsm ad tie ain's, ee apey Owspoun'ssth BLond. hig hof be ta
slllonge allds: fay aly war fefleat haly

othorarin hecately conthremesilo chansprly se Ce semasan or vysomatst h tat Lommed ad w thedlourleed cond
plart, todaf sap:
Maplldemis owansh thengon. anesid d

Ulysses (longer training sequence)

Duke leasned of special pocket of
colours in a thief, were touched, corners for him with sidelope henderprooner
tannally arranged. She had forward, flocking epermonts at quest,
triled, ruted with graceful, habits, following the prophery ectoes with
strong wax.

Ulysses (large batch size):
Dukewslowly. As if he befores. God. By the guys,
I have a minitele time. Lew your mother. Drop inside.

—Parlour of a geose near too, Mr Gusdox Englishman, thigh I wonder. The monkeys I murder
that didn't were his God's lair when I will do that look like a prise over to through
mile. Just canoa. Turns kere the verificers.

IV. Conclusion and further improvements

As can be seen from the generated text, the model is able to preserve some of the style of the text it was trained on, but not the coherence or grammar. We can say that this character-based RNN model generalizes to works from various classic authors, but that, as with the Shakespeare model, the actual content of the text is meaningless. After changing parameters, we found the only one that decreased loss was adding more epochs, which was guaranteed to decrease loss. It is not obvious, however, that the model with more training time produced significantly better text in terms of style replication. Changing the activation function led to a steep increase in loss, and the result contains very few real words. Changing the architecture of the hidden layer to an LSTM also increased loss, and there were almost no real words. It is clear visually that the switch to an LSTM model was the worst of the changes we made in terms of text replication, but the loss was only moderately larger. The superior performance of GRU over LSTM, besides potential issues in optimizing initial parameters differently between RNN's, may be attributable to the fact that the training set was both small, having only been trained on one novel, as well as lengthy, given that Ulysses is composed of over one million characters [7]. There was also a slight increase in loss when using longer sequences and larger batch sizes while no detectable change in the quality of the generated text.

The primary objective of this model is to generate novel text in the style of a particular author, and based on our reading of the output we were able to generate, the model was successful, however we were unable to consult a domain expert to verify this. Specifically, the model may be overfit to the training text, and the stylistic similarities are due to pure replication of the training data rather than simulating novel output. A potential way to rectify may be to check how often long strings of characters are repeated exactly from the training data, although that approach is not foolproof.

Appendix

References

[1] Andrej Karpathy, *The Unreasonable Effectiveness of Recurrent Neural Networks*, http://karpathy.github.io/2015/05/21/rnn-effectiveness/

[2] Tensorflow Tutorials, *Text Generation with an RNN*, https://www.tensorflow.org/text/tutorials/text_generation

[3] Dipti Pawade, Avani Sakhapara, Mansi Jain, Neha Jain, Krushi Gada, "Story Scrambler - Automatic Text Generation Using Word Level RNN-LSTM", International Journal of Information Technology and Computer Science(IJITCS), Vol.10, No.6, pp.44-53, 2018. DOI: 10.5815/ijitcs.2018.06.05 https://www.mecs-press.net/ijitcs/ijitcs-v10-n6/IJITCS-V10-N6-5.pdf

[4] Jason Brownlee, *Text Generation With LSTM Recurrent Neural Networks in Python with Keras,* https://machinelearningmastery.com/text-generation-lstm-recurrent-neural-networks-python-keras/

[5] Shipra Saxena, Introduction to Gated Recurrent Unit (GRU), https://www.analyticsvidhya.com/blog/2021/03/introduction-to-gated-recurrent-unit-gru/

[6] Abigail See, Natural Language Processing with Deep Learning: Vanishing Gradients and Fancy RNN's, https://web.stanford.edu/class/cs224n/slides/cs224n-2019-lecture07-fancy-rnn.pdf

[7] Yang, Shudong & Yu, Xueying & Zhou, Ying. LSTM and GRU Neural Network Performance Comparison Study: Taking Yelp Review Dataset as an Example. 98-101. 10.1109/IWECAI50956.2020.00027, https://www.researchgate.net/publication/347267378_LSTM_and_GRU_Neural_Network_Performance_Comparison_Study_Taking_Yelp_Review_Dataset_as_an_Example/citation/download

[8] Chung, Junyoung, et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling." *arXiv preprint arXiv:1412.3555* . https://arxiv.org/pdf/1412.3555v1.pdf

Dataset:

Ulysses: https://www.gutenberg.org/files/4300/4300-0.txt
Dubliners: https://www.gutenberg.org/files/2814/2814-0.txt
The Iliad: http://classics.mit.edu/Homer/iliad.mb.txt
The Odyssey: http://classics.mit.edu/Homer/odyssey.mb.txt

Group member contributions:

Jacky: Trained the model on *Ulysses*, *The Iliad*, and *Dubliners.* Experimented with training the model on Ulysses with different batch sizes and length of training sequences.
William: Selected the works to train the models, reviewed literature discussing the effects of changing parameters, contributed mainly to writing the introduction, methods, and conclusion.
Rebecca: Reviewed and compared literature discussing changing parameters, contributed findings to conclusion, methods, and results.
Rehan: Trained the RNN on the *Odyssey* dataset, in addition to training the model on three other conditions: a greater number of epochs, a tanh recurrent activation function, and a LSTM rather than GRU layer. Contributed mainly to the results portion of the paper.

Code, trained models, and datasets:

https://github.com/rehanbchinoy/Math-156-project