Recurrent Neural Network: Exploring Variants for Character-Level Text Generation

Giovanni Mantovani and Andrea Palmieri

May 12, 2024

1 Introduction

The objective of this study is to explore the effectiveness of various Recurrent Neural Network (RNN) configurations in the context of character-level text generation. Focusing on different RNN architectures such as LSTM, GRU, and traditional RNNs, this study investigates the impact of modifications in the number of layers and hidden state dimension on the model's performance. By analyzing these models using the "Alice's Adventures in Wonderland" text as a dataset, our goal is to identify which configurations yield the most coherent and contextually appropriate text generation.

2 Methodology

2.1 Dataset

The dataset utilized for the training of the model is derived from the book *Alice's Adventures in Wonderland* by Lewis Carroll, available at https://www.gutenberg.org/ebooks/11.txt.utf-8. This text-based dataset includes the complete narrative of the novel, encoded in UTF-8 format. It contains all chapters of the original work, as well as a preface, totaling 163,916 characters.

For data preprocessing, the text is converted to lowercase and non-alphanumeric characters are removed, leaving only letters and spaces. The dataset is then partitioned into a training set covering 70% (excluding the initial 10% segment) of the text and a testing set comprising the remaining 20%.

- `Batch size` = 256.
- `Loss function` = CrossEntropyLoss. Essential for calculating the loss by comparing the predicted probability distributions (output logits from the model) with the actual target character labels.
- `Optimizer` = Adam. Chosen for its effectiveness in handling sparse gradients and noisy problems, common in natural language processing tasks.
- `Dropout rate` = 50%. Applied within the recurrent layers to help mitigate overfitting.
- `Sequence length` = 100. This defines the length of input sequences, setting the number of characters the network uses at once to make predictions.
- `Hidden units` = 256 and 512. It's the size of the hidden state in each layer, affecting the network's capacity to learn and remember information over sequences.
- `Number of layers` = 5. This defines the depth of the RNN, which plays a crucial role in the network's ability to model complex patterns within the data.

2.2 Models Used for Testing

We compare several recurrent neural network (RNN) models, including different types such as RNN, GRU, and LSTM, to assess the impact of architectural variations and parameter settings on text generation performance. These configurations are designed to explore the influence of layer counts, hidden unit sizes, dropout rates, and advanced features like layer normalization.

- **Model 1:** GRU with 5 layers, 256 hidden units, and 0.5 dropout.

- **Model 2:** LSTM with 5 layers, 256 hidden units, and 0.5 dropout.
- **Model 3:** LSTM with 5 layers, 512 hidden units, and 0.5 dropout.
- **Model 4:** GRU with 5 layers, 512 hidden units, and 0.5 dropout.
- **Model 5:** RNN with 5 layers, 256 hidden units, and 0.5 dropout.
- **Model 6:** RNN with 5 layers, 512 hidden units, and 0.5 dropout.

2.3 Training

The training was set up with a total of 10 epochs and a batch size of 256. Training data was shuffled for each epoch to prevent the model from learning any order-dependent patterns in the training set, thereby improving the model's ability to generalize. During testing, the batch size was instead set to 1. This is particularly important for character-by-character prediction, as it provides a granular view of the model's performance at each step. To prevent overfitting and ensure continued improvement, an early stopping mechanism was employed. The training loop included checks for validation loss improvement; if no improvement was observed after six consecutive epochs, the training would halt. The model saved at the point of lowest validation loss is considered the final output.

3 Results

3.1 Comparison of Text Generation at Different Temperature Settings

The temperature value controls the randomness of predictions by scaling the logits before applying softmax. This section explores the impact of different temperature settings on the coherence, creativity, and readability of the text generated. The following list outlines the effects observed at various temperature settings:

- **Temperature 0.5:** At this conservative setting, the model produces text that is coherent but closely aligned with the structure and style of the training data. However, the results vary significantly depending on the network architecture.
- **Temperature 0.7:** A slight increase in temperature introduces greater diversity in word choice and structure, along with a noticeable increase in spelling errors.
- **Temperature 1:** Misalignment with the provided input increases at this setting, and incorrect words begin to predominate.
- **Temperature 1.5 and 2:** At these higher settings, the model's outputs exhibit a high degree of randomness, resulting in text that completely lacks logical structure and coherence. The words are almost all spelled incorrectly.

3.1.1 Comparative Examples

Prompt: "did not at all know whether it would like the name however it only grinned a little wider come its p"

- **Temperature 0.5:** "n the went on eeeiting alice seplied to be for so she caterpillar i shall alice was ier in a srile the shought and begond it was a mittle but she was wou did not looked to and she ootpe with it out to alice iad bestainly as loue she qooal of the lock turtle shought so all h can be to all more and the was lorely alice but it would gor to the caterpillar she konk and luch and the was monged it was to alice who was to she oant in and raid the dormouse with the sabbit cried she baby"
- **Temperature 0.7:** "e whem in the rween and wonder it it what j thould i dan dvighted the hatter wou ho the was quite so said the sane with of course began begind it is was co she qioe in whe oous had at the rame of the sabbit caml all the was a largg darch said alice she shought all the sight began time without bome wourhlg about her alice hime or to the winld whth it was a she lock turtle temlied to the would "

- **Temperature 1:** "it the tent bu yould nake lawtily we cr his ark many layt ate do so alice lunking but bomd i shinking of the came buoing all adver her at soeez the soomd shouihng aftted said alice but sheye was mott fas had a hine thoimk i haves eollw doary when begin rhat she iury butie it vas gou eealt uay duulded off and all al ereduiou crobpued uhe putdtely for in fe rhe midhcale sonn a little bould bome of truting a oowhtiite frolau tp swo in the ceak rhe wselte but he eid poler he said the oot out"
- **Temperature 1.5:** "ettiat dyt handss exerytwnnt yalky mor leptis femuhsplng wou ne a cnumd not shink magus wau nf shiw wlle clolm vr asnisy suertinn as see wwing diiatdd eysl jn iv no lessanly teol y iutoage so metsaro what prcer putnyitiiiph a dhandrmtptme uirh sonaaberal a vkalina tsiog as oed attiv abeecrhng lyrt said alice wery suidf ay vhesheo being fow defae down all ornoopvmdsilylys ard gl the eigqoal adou vit fathertlit the way she suuzal clo su see taid alice cotestwaninged toslcied at lyrtionsjng ha"
- **Temperature 2:** "ith ort errsvaredt olppyrpbsaitnwh you drpiows teey thoulddned ip a pouhmging mauoisgly oay suori aevcodefrentted i moth wru curidteloled the askenper cveanu jm oystoihty-isyoyoo pwite dxeavurinuisy for eo lip fiodkveddd awsyieyno fisiableqs policedaol ieew back this cat oly wat gntorlboe to goad a eouvraa qt fu itst rheys wiunfe ieeted adyarmfd imtoiands ve-neapditielclabll onbbuifhc bhtwlsw it dacied nunling abhice tas thrnggt suupitgands sp spd mloke chirihnylog ho a hell upoptiting amice"

4 Interpretation of results

In this section, we analyze the outputs of the trained models under a consistent setting, using a fixed temperature value of 0.5. This approach allows us to directly compare the performance and text generation capabilities of each model configuration.

- **RNN Model (5 layers, 256 hidden units, 0.5 dropout)**: The basic RNN struggles significantly with coherence and relevance, producing many outputs that are disjointed and lack meaningful connection to the prompts. This model demonstrates the limitations of simpler RNN architectures in handling complex sequence generation tasks.
- **RNN Model (5 layers, 512 hidden units, 0.5 dropout)**: Although an increase in hidden units provides some improvement, the basic RNN architecture still underperforms compared to GRU and LSTM models. The text generated is often repetitive, prone to grammatical errors, and strays from the prompt more frequently than its more sophisticated counterparts.
- **GRU Model (5 layers, 256 hidden units, 0.5 dropout, 12 epochs)**: This model configuration benefits from the inherent stability and efficiency of GRU architectures, but it shows mixed performance in generating coherent text. The results demonstrate variability in coherence and grammatical correctness, with performance notably fluctuating based on the complexity of the prompts.
- **GRU Model (5 layers, 512 hidden units, 0.5 dropout, 10 epochs)**: More similar to the LSTM with the same hidden unit configuration, this GRU model also benefits from increased model capacity, delivering text that is relatively coherent and more aligned with the prompts. It competes closely with the LSTM model, although it occasionally produces slightly less consistent narrative connections.
- **LSTM Model (5 layers, 256 hidden units, 0.5 dropout, 11 epochs)**: This LSTM configuration shows improved coherence over the GRU model, with fewer repetitions and a stronger adherence to the given prompts. The generated text demonstrates a better grasp of "narrative flow", although there are occasional nonsensical phrases and some grammatical errors. The LSTM model appears more robust in handling longer sequence predictions and maintaining context.
- **LSTM Model (5 layers, 512 hidden units, 0.5 dropout, 10 epochs)**: Increasing the number of hidden units to 512 has further refined the output of the LSTM model. The text generated is noticeably more coherent and contextually appropriate, with richer vocabulary and more complex sentence structures. This model shows the best performance in terms of generating plausible, engaging text that follows the logical progression from the prompts.

5 Conclusion

The results of this study demonstrate that LSTM and GRU models, particularly those with higher numbers of hidden units, consistently outperformed the basic RNN configurations in terms of coherence, context relevance, and word correctness. LSTM models with increased hidden units (512) resulted in the highest efficiency, producing text that was not only more coherent but also rich in vocabulary and syntactically complex. These models were adequate at maintaining logical progression from given prompts, suggesting capacity for longer sequence predictions and superior handling of context. GRU models displayed similar capabilities, proving to be competitive with LSTMs, though with discrepancies in narrative continuity. Finally, the basic RNN models struggled with the task, highlighting their limitations in maintaining coherence over extended sequences.