Project #4: Recurrent Neural Networks with Tensorflow and Keras

Amir Sadovnik
COSC 525: Deep Learning (Spring 2021)

---

1 Overview

In this project you will be building a character based recurrent neural network (RNN's) to write Beatles songs. We will frame the problem as a many-to-one task in which given a certain number of characters your goal is to predict the next character. You will experiment with different RNN cells and different parameters.

2 Dataset

You will be using a text file which includes lyrics from 246 Beatles songs. All the lyrics are concatenated to each other and will be treated as one long sequence. The text file is attached to the assignment on Canvas. The lyrics are taken from the following website:

http://beatlesnumber9.com/lyrics.html

3 Problem Description

You are tasked with writing the following functions:

1. Write a method which given a text file name, a window size and a stride it creates the training data to perform back propagation. This can then be saved in a new text file for later training where each line is a sequence of length $window\_size + 1$ (thus giving the predicted character given the first $window\_size$ ones). For example:

```
Original text: hello, how are you?
Window Size: 5
Stride Size: 3
Output file:
```

```
hello,
lo, ho
 how a
w are
re you
```

2. Write a method which given a file name in which each line is a single training sequence, returns the input and output array for training. More specifically, the input x should be an array of size $m \times n \times p$, where m is the number of sequences, n is the length of the sequence and p is the vocabulary size (in order to do 1-hot encoding). The output y should be an array of size $m \times p$ which is the predicted output. This will be the data you use for training.

3. Write a method which predicts a given number of characters given a certain model and *window_size* characters to initialize. The method should accept the initial characters, the model, the sampling temperature ($q_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}$), and the number of characters to produce and then iterate to create that number of characters.

4. Write a method to perform training on a specific model. The method should accept the model (you will try different ones as described next), the training data, number of epochs, and any other learning parameters you choose. Every certain number of epochs, generate a few sequences by initializing them with random samples from your training data and generating multiple characters.

5. You are asked to compare SimpleRNN and LSTM, each with at least two configurations (size of hidden state) and each with at least two window size and strides. This should give you 8 total models to compare. **Optional:** Use a mulitlayered rnn.

6. Compare both the loss-epoch for all 8 configurations in addition to quantitative results by generating some text (you can initialize this with text that is not in your dataset.

4 Additional Information

1. Exact learning parameters are not mentioned. You will need to select your own learning rate, momentum etc.

2. You should have a set of commend line variables which allow the user to run each of the configurations. For example:

   ```
   python3 rnn.py beatles.txt lstm 100 10 5 1
   ```

   Will run the code with an LSTM with a hidden state of size 100 and a window size of 10, a stride of 5 and a sampling temperature of 1.

5 Report

You should submit a short PDF report with the following:

1. A short introduction to the problem.
2. Description of your different networks including learning parameters.
3. Results from each network. This should include the loss plots and the generated text (try to generate a few lines of text).
4. Conclusion
5. How to run your code.

6 Submission

You are required to submit one zip file with the code and your report.