Foundations of AI
Instructor: Rasika Bhalerao
Assignment 5
Due June 18

This assignment builds upon Homework 1 (the character ngram language model). We will update the weights in the model using reinforcement learning to train the model to generate text that fits a specified criteria. I recommend making a copy of your solution to Homework 1 in a new notebook, and modifying the code in there.

Learning goals:

- Implement Q learning
- Analyze a Markov Decision Process

What to do:

1. I recommend using a dataset where each document is small.
    a. Here is an example dataset of Haikus: https://www.kaggle.com/datasets/bfbarry/haiku-dataset
    b. Here is an example dataset of Olivia Rodrigo's songs: https://www.kaggle.com/datasets/mehaksingal/olivia-rodrigo-lyrics-datasetl
2. In the following steps, we will be updating the weights in CharNGramLanguageModel using Q learning. This may make some of the weights negative, which will not work with the current version of the Homework 1 code. We can accommodate negative weights in several ways (listed below). Please try both, but leave only one version in the code that you submit. There are questions to answer about both. When building the dataset of counts of each next letter, we will not divide by the total to convert it to a percentage. Instead, we keep the values as the whole number counts. Then, in the generation stage, we can convert the weights to positive numbers by:
    a. Subtracting the minimum and adding 1 to each number
    b. Passing the list of weights through the softmax function
        i. You may `from scipy.special import softmax`
3. Create a class called `ReinforcementLearning`.
    a. Its constructor should take three arguments: a `CharNGramLanguageModel`, and two numbers for `alpha` and `gamma`. It should store all three to instance variables.
    b. Add a method called `Q_learn(self, criteria, num_prompts = 1, iterations_per_prompt = 30)`.
        i. `criteria` is a function which takes a string as the argument and returns a numerical score, which will be used as the reward for Q learning. The client can use this argument as a way to specify which kind of text they prefer. For example, they can pass this function as the `criteria`

argument to give higher rewards to shorter text: `lambda x: -len(x)`. They could also pass the `predict()` function of a trained binary classifier, or a function which just prints the text and asks the user to specify (on a numerical scale) how much they like the text.

ii. `num_prompts` is the number of times that this method should ask the user for a prompt in order to generate text.

iii. For each prompt, this method should use the CharNGramLanguageModel (from the constructor) to generate text `iterations_per_prompt` number of times. Each time, it should pass the generated text to the `criteria` function and use the resulting score to update the weights in the current CharNGramLanguageModel using Q learning (with the `alpha` and `gamma` passed to the constructor).

4. Test out whether the model is now more likely to generate text that fits the specified criteria!
    a. Create a new instance of a `CharNGramLanguageModel`.
    b. Use that model to generate 10 pieces of text (using the same prompt for all 10) and print out the average length of the generated texts.
        i. You may also print out a sample generated text.
    c. Create a new instance of `ReinforcementLearning`. Pass the `CharNGramLanguageModel` as an argument to the constructor.
    d. Call the `Q_learn()` method.
        i. For the `criteria` function, test out `lambda x: -len(x)`
    e. Use the same `CharNGramLanguageModel` to generate 10 more pieces of text (using the same prompt for all 10) and print out the average length of the generated texts.
        i. You may also print out a sample generated text.

What to turn in:

Please submit these via Gradescope:

- Your Python code
- A text or pdf file with your answers to these questions:
    - Questions specific to this assignment:
        - With a criteria function of `lambda x: -len(x)`, did it encourage the model to generate shorter texts? Why or why not?
        - Try out a criteria function of your choice. You could pass the `predict()` function of a trained binary classifier, or a function which just prints the text and asks you to specify (on a numerical scale) how much you like the text. When testing it out, in this version of steps 4b and 4e, instead of printing the average length, you'll need to print the average score output by this criteria function. Are you able to encourage the model to generate texts in your preferred style?

            - If you choose to manually provide your opinion on texts, try to stay consistent with your preferences so it is possible to evaluate whether the model was encouraged to generate text in your preferred style.
        - In Homework 1, the character ngram language model was using frequencies as the weights when choosing the next letter, but for Homework 5, we switched it to accommodate negative values as a result of Q learning. One option we tried for this was to use the softmax of the weights, which resulted in the generated songs becoming longer. Why?
        - We also tried subtracting the minimum and adding 1 to each number. Why did we add 1 instead of just subtracting the minimum?
        - In the generation stage, the difference in the length of generated songs (as a result of Q learning) is much more dramatic if we test the same prompt before and after Q learning. Why is the difference more dramatic when we test the same prompt, as compared to using a different prompt before and after Q learning?
    - The usual questions:
        - How long did this assignment take you? (1 sentence)
        - Whom did you work with, and how? (1 sentence each)
            - Discussing the assignment with others is encouraged, as long as you don't share the code.
        - Which resources did you use? (1 sentence each)
            - For each, please list the URL and a brief description of how it was useful.
        - A few sentences about:
            - What was the most difficult part of the assignment?
            - What was the most rewarding part of the assignment?
            - What did you learn doing the assignment?
            - Constructive and actionable suggestions for improving assignments, office hours, and class time are always welcome.