Foundations of AI
Yunyu Guo
Assignment 4

- 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?

    In the reward function in the q_learn method, the function: lambda x: -len(x) is designed to reward shorter texts. This is because the reward is negative of the length, so shorter texts will have higher (less negative) rewards.

```
BEFORE: Please enter the first word(s): stranger
47.7
stranger I know everything about

AFTER: Please enter the first word(s): stranger
42.6
'stranger I knows that they believe it's nice\n'
```

  - 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.

      I add 5 to the count when any of these words: happy, nice, love occurs. The rewarded count is then used as reward in the q function.

```
for word, count in word_counts.items():
                if next_letter in window_freq:
                    window_freq[next_letter] = (1 - self.alpha) * window_freq[next_letter] + self.alpha * (count +
self.gamma * max(window_freq.values()))
                else:
                    window_freq[next_letter] = self.alpha * (count + self.gamma * max(window_freq.values(),
default=0))
```

The model was able to generate text includes one of those words once in 10 times.

```
BEFORE: Please enter the first word(s): stranger
stranger I know everythings I did to try and win your loved you can't even remember what made me lose all the day that I am the storm cl

AFTER: Please enter the first word(s): stranger
stranger I know

stranger I know everybody told me it would burn out of my life

stranger I know ever doin' all my sleep

stranger I knows that it doesn't matter anymore, no

stranger I know

stranger I know ever keep so far out of all the storm clouds would burn out and it would burn out of my life

stranger I know ever doin' and now I'll love

stranger I knows that they believed that I'll everything about, ooh, ooh, ooh

stranger I know everythings I did to go

stranger I know
```

- 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?

  The softmax function converts a vector of values into a probability distribution. It does this by exponentiating each value and then normalizing by the sum of all exponentiated values.

  Because of the exponentiation effect, the softmax function amplifies differences between values. This can lead to a more skewed probability distribution where certain characters (those with higher weights) are much more likely to be chosen, it will be more likely to generate those sequences repeatedly, leading to longer texts.

```
BEFORE: Please enter the first word(s): stranger
50.1
stranger I know everybody told me it was half myself without you, I'll everything about

AFTER: Please enter the first word(s): stranger
64.9
'stranger I know\n'
```

- We also tried subtracting the minimum and adding 1 to each number. Why did we add 1 instead of just subtracting the minimum?

  By adding 1 after subtracting the minimum, all values will be strictly positive (greater than zero).

- 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?

  Using the same prompt provides a controlled comparison, allowing the effects of the learning process to be more clearly observed.

  The learning process may have reinforced certain patterns or transitions that are particularly relevant to that specific prompt.

- The usual questions:
  - How long did this assignment take you? (1 sentence)
    3 days
  - 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.

      I discuss with a student about how the reward (-len(x)) should be used as to consider from the first generated letter to the current letter (one letter after the window). Also what other criteria or reward can be designed.

  - Which resources did you use? (1 sentence each)
    - For each, please list the URL and a brief description of how it was useful.

      Course slides
      Olivia Rodrigo's songs:
      https://www.kaggle.com/datasets/mehaksingal/olivia-rodrigo-lyrics-datasetl
      I use the lyric "stranger" as training text.

  - A few sentences about:
    - What was the most difficult part of the assignment?

The reward or criteria should consider from the fist character of the generated text to the window + next letter. Also should pay attention to when to convert negative weight, it should convert the negative weight after each iteration in Q learning (total 30 iterations).

When design another criteria function such as ask user to enter a score of a generated text, it seems difficult since the reward takes into account window plus next generated character, which makes it frequently ask user to enter a score when window sliding in order to update the score. Because the q learning should update the weight in each letter generation, not update the weight after the whole text generated.

- What was the most rewarding part of the assignment?

  Able to try using the q learning model and compare with the ngram model.

- What did you learn doing the assignment?

  How to design reward based on how the weight in the ngram model is calculated. It should be count relative and easy to update when moving the window.

- Constructive and actionable suggestions for improving assignments, office hours, and class time are always welcome.