Gradient descent

Start with the provided `unigram_pytorch.py`.

1. Choose a `num_iterations`.
2. Choose a `learning_rate`.
3. Augment the file to build visualizations of:
    1. **the loss as a function of time/iteration - also include the (known) minimum possible loss**
    2. **the final token probabilities - compare this to the (known) optimal probabilities**

Tweak your `num_iterations` and `learning_rate` to get reasonably good results reasonably quickly (seconds).

You should turn in a document (`.txt`, `.md`, or `.pdf`) answering all of the **red** items above. You should also turn in your modified `unigram_pytorch.py`. Unless otherwise specified, you may use only `numpy`, `matplotlib`, and the standard library.