

Gradient descent

Start with the provided `unigram_pytorch.py`.

Explain what this neural network does. What are the inputs and outputs of the learned function?

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

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

Describe how you could modify/augment this code to perform document classification.

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` (the **blue** items). Unless otherwise specified, you may use only `numpy`, `matplotlib`, and the `standard library`.