Su Zhang
Gradient Descent Explanations

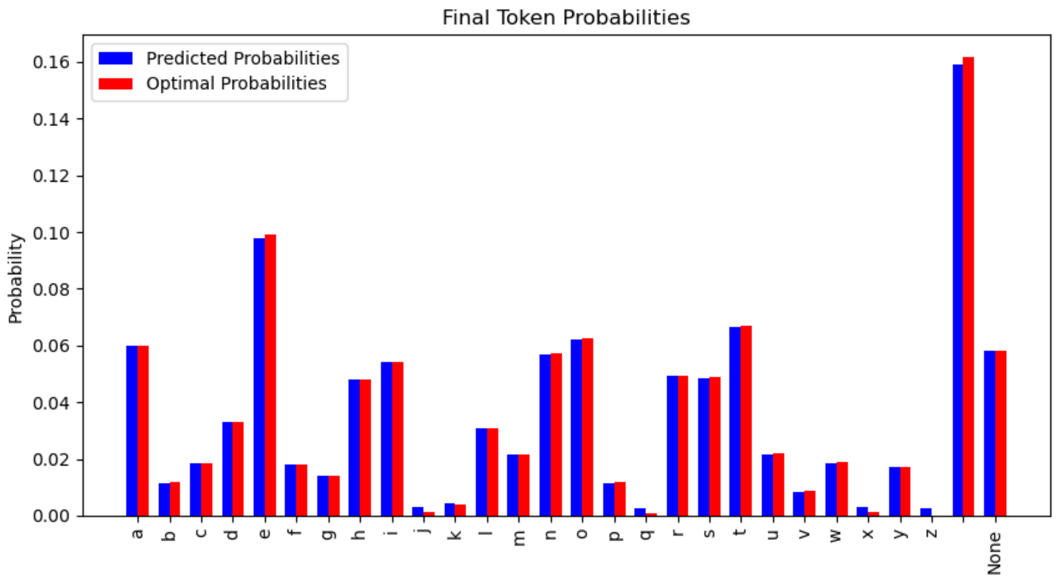
Explain what this neural network does. What are the inputs and outputs of the learned function?

This neural network primarily trains unigram model to predict the probability distribution of the tokens in a given text. The neural network aims to approximate the probability of each token in the vocabulary through adjusting parameters to maximize the probability of observing the tokens in the given text.

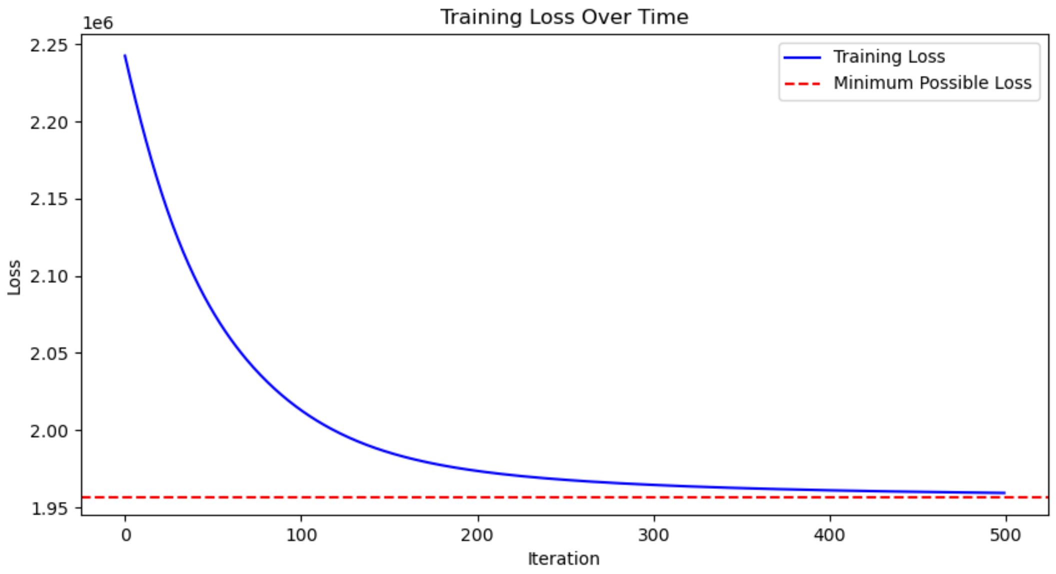
The learned function is "unigram" function. The **inputs** of the learned function are one-hot encoded vectors of tokens from the vocabulary and **outputs** are the sum of the log-probabilities of each token in the document.

Data visualizations:

Graph 1: The final token probabilities - compare this to the optimal probabilities (when number of iterations is 500 and learning rate is 0.01)



Graph 2: The loss as a function of time/iteration - also include the minimum possible loss based on the optimal probabilities (when number of iterations is 500 and learning rate is 0.01)



When number of iterations is 500 and learning rate is 0.01, the training time is 1.7623 seconds, which achieves reasonably good results with reasonably fast speed.

```
Training time: 1.7623 seconds
```

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

To perform document classification, the neural network should be able to map given text / document into one of a set of discrete classes. For example, in authorship attribution, the neural network model is trained to maximize the probability assigned to the correct author. For this specific code, I would do the following to perform document classification:

- Instead of treating each character as tokens, I would make each unique word as tokens and remove all the punctuations or spaces from the training documents
- The training datasets should also include multiple documents with different authors
- Find suitable ways to extract features from documents, such as "Bag of Words" or "Term Frequency-Inverse Document Frequency"
- Assume that we are using linear classifiers such as naïve Bayes for the classification, we should update the formula in the learned function to calculate the product of prior probability of the class P(class) and the likelihood of the document P(document|class)
- After updating the learned function, I would then update the loss function that is suitable for classification problems, such as categorical cross-entropy loss