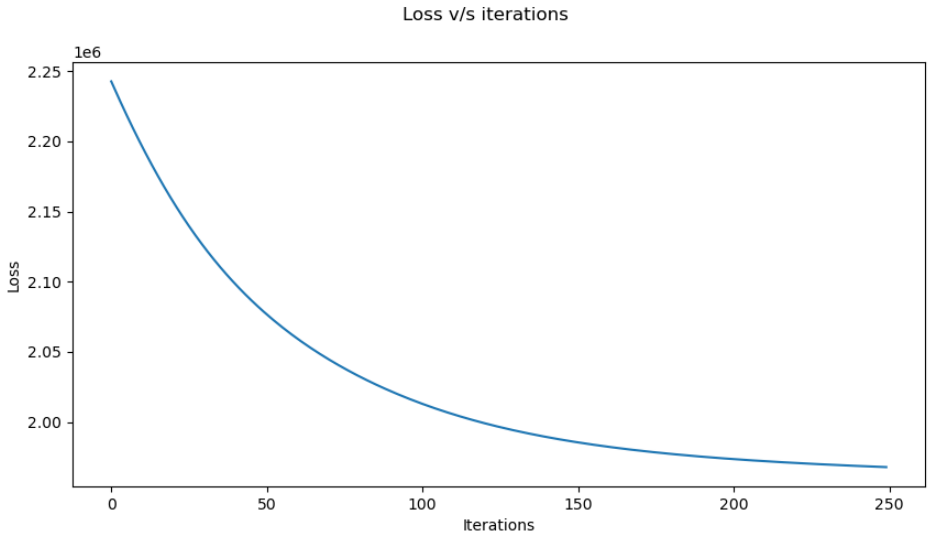
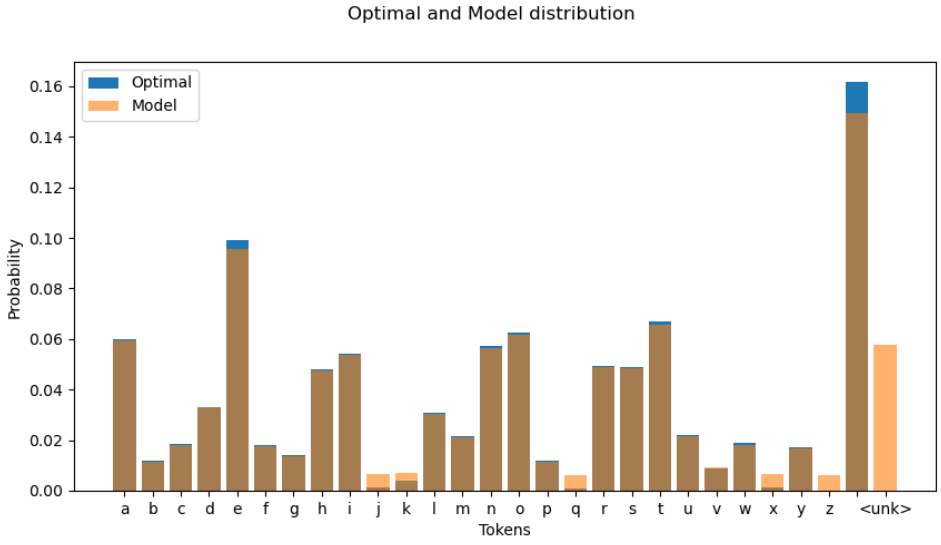
Natural Language Processing Assignment: Gradient Descent

The neural network defined in the python script tries to learn the probability distribution function of the input training vocabulary by modeling it as a unigram distribution. The parameters of the neural network are the parameters of the categorical probability distribution of the tokens in the vocabulary.

The optimal probability distribution is = (#Occurrences of the tokens in the vocabulary)/(# Total tokens in the vocabulary)

With num_iterations = 250 and learning_rate = 0.01





The time taken to converge is : 3.94s

Modifying the code for document classification:

1. We could train different neural networks, each with a different set of parameters, for each class. So if we have n classes in our dataset, we would train n neural networks, learning n*len(vocabulary) number of parameters to describe $p(x|c_n)$. We can then find the total probability of the document (all the words x_i) belonging to class c_n as
   $p(\{x_i\}\ belongs\ to\ c_n) = p(c_n)\prod_i p(x_i|c_n)$ and then choose the class which has the highest probability
2. Alternatively, we could create a multi-layered neural network, with the first layer having len(vocabulary) parameters, the second layer having n parameters, where n is the number of classes. We then train the parameters to optimize the cross-entropy loss of the training distribution class assignemnt, i.e., p(c|xt) where xt is a training sample.