Statistical Approaches to NLP
PA 2 – MaxEnt Writeup

José Andrés Molina

October 27, 2016

1 INTRO

The chosen features chosen were the 750 most common words in the training set. In the Document's features() function, the data was tokenized and stopwords and punctuation were removed. The list of remaining tokens was then returned. In the MaxEnt classifier's train() function, a set of all features was collected and trimmed to the most common 750 using a Counter. This outperformed most common 100 and most common 500 significantly, but was still reasonably fast, when compared to using 10,000 or the whole feature set (neither of which gave much of a significant performance improvement anyway).

The learning rate used was 0.001. This didn't improve accuracy, but it did help the model converge more quickly, improving performance in terms of speed over 0.0001.

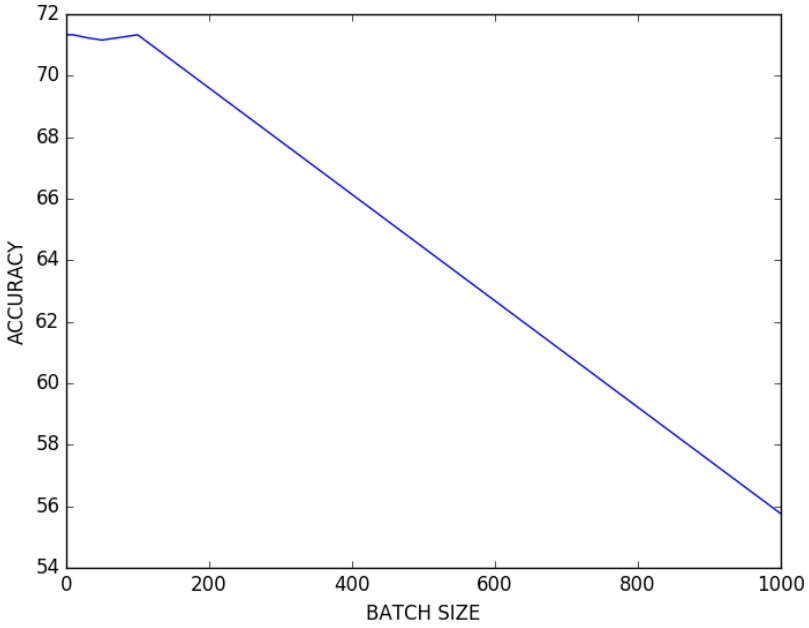
I chose to use accuracy to determine when converged and begin early stopping after a patience of 5. Using negative loglikelihood only improved the final training set accuracy by a fraction of a percent for most of the experiment cases, by letting it run for longer to find a better maximum, but it could run for over 100 epochs and was much, much slower than stopping it much earlier with accuracy over the dev set. This was ruled out as not worth the time for the minimal improvement.

The final accuracies with a batch size of 100 and a learning rate of 0.001 (and a training set size of 10,000 for the Yelp reviews) are 76.38% on the Names corpus (gender identification) and 71.33% on the Yelp reviews (sentiment analysis for "positive", "negative", "neutral" reviews).

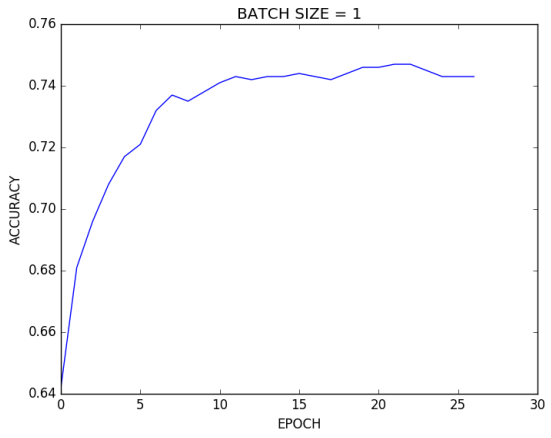
2 EXPERIMENT 1: BATCH SIZE

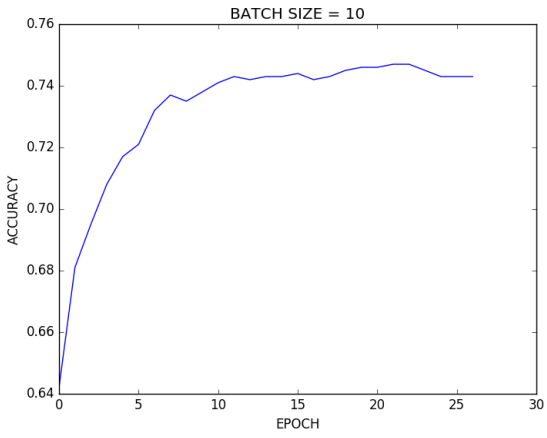
Regardless of the batch size, the accuracy on the test set didn't change much at all for the smaller batch sizes of 1–100. But at a batch size of 1000, it was much much lower possibly because it was stopping too early for that massive batch size.

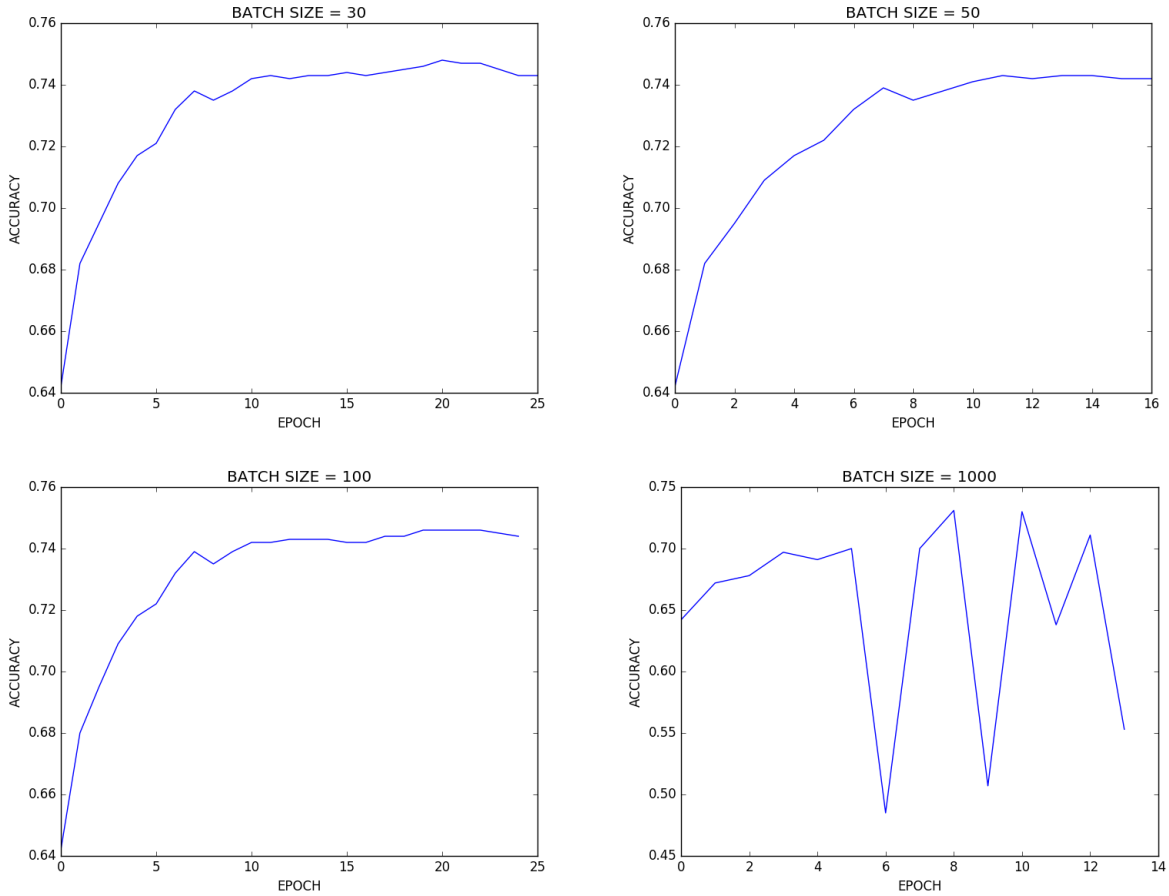
These were all run with a training set size of 1000, a learning rate of 0.001, and using the most common 750 words in the training set as features (after removing stopwords and punctuation).



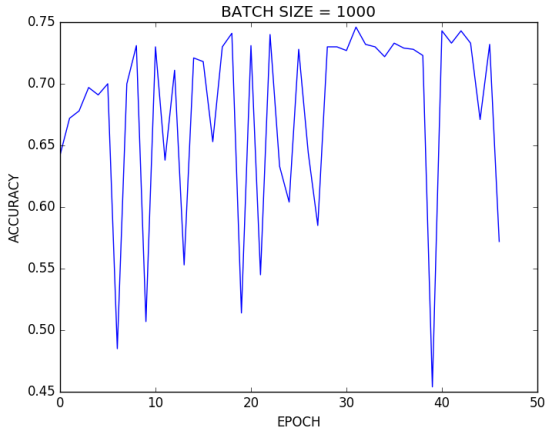
Below are all the the convergence curves for each batch size over the epochs. As shown in the one for 1000, it does not look like it is anywhere near convergence when it exits the loop. (Accuracies in this case are on the dev set.)







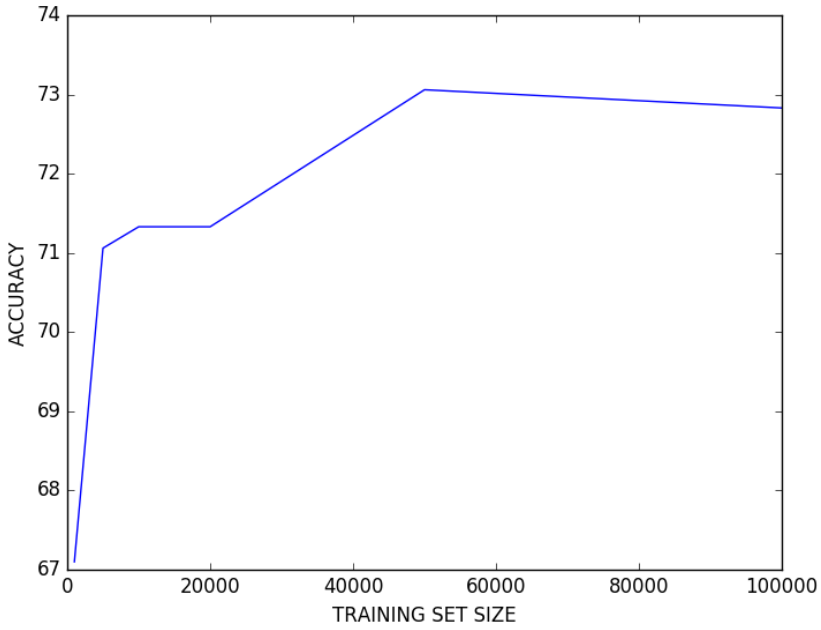
As a result I reran the batch_size=1000 experiment a second time with a patience of 15 epochs without change instead. It ran for much longer (for many more epochs), but still did not show any signs of converging properly. Although the final training set accuracy was a slightly better 57.2% over the 55.76% with the original patience of 5.



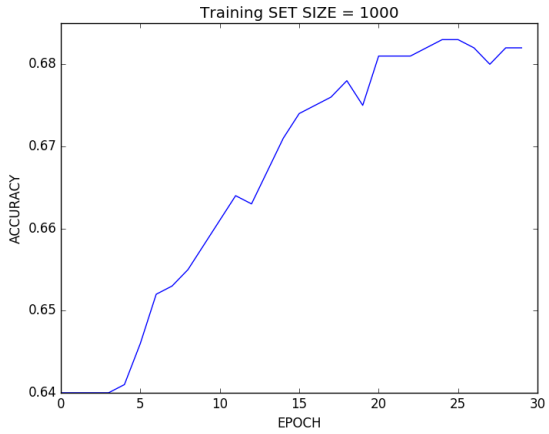
3 EXPERIMENT 2: TRAINING SET SIZE

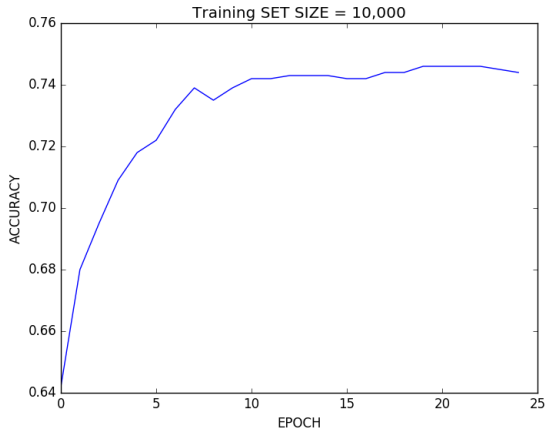
Increasing the training set size does seem to impact the the accuracy on the test set, but it seems to peek at 50,000 with 73.06% accuracy, doing slightly worse at 100,000 with 72.83% accuracy.

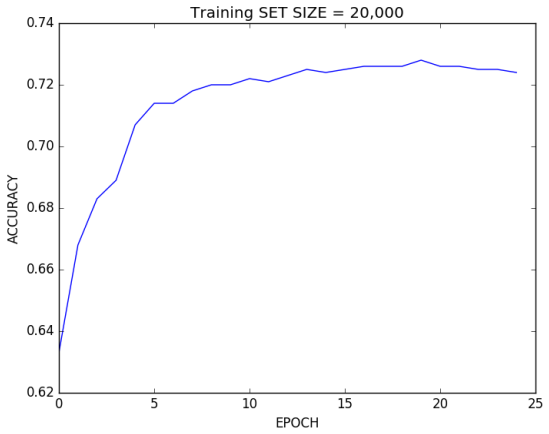
These were all run with a batch size of 100, a learning rate of 0.001, and using the most common 750 words in the training set as features (after removing stopwords and punctuation).

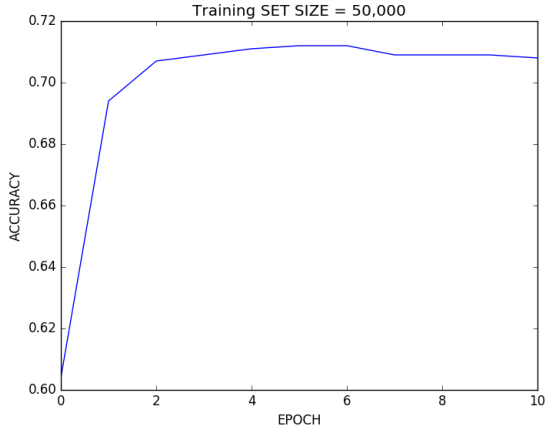


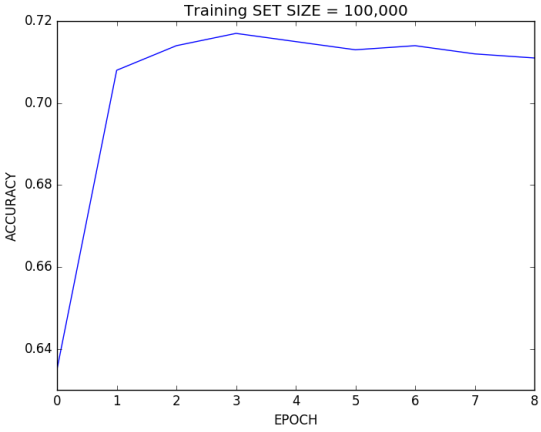
Below are all the the convergence curves for each training set size over the epochs. Notice also that it converges in far fewer epochs as the training set gets larger (however, each epoch is far slower, so time-wise it's still far slower).




Training SET SIZE = 10,000
ACCURACY
EPOCH
0.76
0.74
0.72
0.70
0.68
0.66
0.64
0
5
10
15
20
25


Training SET SIZE = 20,000
ACCURACY
EPOCH
0.74
0.72
0.70
0.68
0.66
0.64
0.62
0
5
10
15
20
25


Training SET SIZE = 50,000
ACCURACY
EPOCH
0.72
0.70
0.68
0.66
0.64
0.62
0.60
0
2
4
6
8
10


Training SET SIZE = 100,000
ACCURACY
EPOCH
0.72
0.70
0.68
0.66
0.64
0
1
2
3
4
5
6
7
8