

# Logistic Regression with Bag of Words Classifier Write Up

Alix Gates

April 27, 2019

## Model

In the model, I was able to implement everything except the DataLoader because I got the `BrokenPipeError: [Errno 32] Broken pipe` and did not have access to the CS Linux development. Before my Linux computer crashed, however, I used the DataLoader function without that error and noticed that passing in batches of 500 performed much faster than passing in one line at a time. However, I noticed that it took longer for my model to get to an accuracy of 80% when passing in batches of 500 compared to passing in line by line. Passing in batches of 500 started the model at an accuracy of 50% and eventually arrived at an accuracy of around 80% by 25 epochs. When passing line by line to the classifier, the accuracy was at or above 80% on the first try.

## Evaluation

The original model I ran used an Adam optimizer, a learning rate of  $1e-3$ , and 3 epochs. With this set up, the best accuracy achieved was 82% after only 2 passes of the data. Further, the model had an accuracy of 81.34% even on the first pass, only increasing a few tenths of a point to get to the accuracy achieved in the second pass.

Now I will discuss the differences seen under the three changes below:

### SGD vs. Adam

The SGD optimizer achieved an accuracy of 78% on the first pass and achieved the highest accuracy of 83% on pass 3 of the data. This indicates that while it took longer to attain, the SGD optimizer did attain a higher accuracy overall.

### Learning rate of 0.01 vs. $1e-3$

Increasing the learning rate to 0.01 increased the accuracy of the model to 83%, similar to changing the optimizer to SGD. That said, the increased learning rate achieved a 82% on the first pass, similar to the original model's second pass, and achieved the 83% on the second pass. The third pass did not yield a higher accuracy.

### 5 Epochs vs. 3 Epochs

While I would have rather compared 3 epochs with 30 or more epochs, my lack of batching would have meant this model would run for over 2 hours and I did not have the time. Comparing 5 Epochs with 3 epochs, I found that the additional 2 epochs added little to the accuracy of my model. After the second epoch, it was at 82.6%, and after the fifth it was 83%. When I did have batching, I saw that more epochs resulted in a higher chance of attaining a model with higher accuracy. Considering the model's accuracy began at 50%, however, the additional epochs were useful. I also found that the classifier would find the highest model at around 25 epochs before plateauing or even decreasing again. The overall trend with batching plus higher epochs

was waves of lower accuracy and higher accuracy. Feeding in line by line, the accuracy starts so high (82%) the additional epochs are nice but do not add a lot to the accuracy of the model. This is especially true for movie review classification. If the issue were medical, the increased accuracy would likely be welcome and necessary. Further, there is a trade off between the immediate high accuracy seen early on by feeding in line by line and the eventual high accuracy seen later on by batching. Overall, I would think batching with more epochs would be a better experiment for the model.

## Testing

Overall, my model's accuracy came in at 83%, which was similar to the percent accuracy seen in the models above.

## Outcomes

Below I have listed the four categories and the top 10 features under each category. It is logical that the top 10 features with max positive weights for positive category and max negative weights for negative category are the same, and the same applies to the other two categories (that the words are the same between the two). This is because the words that are most associated with a positive category will also be the words that are least associated with the negative stories. Four oddities stand out to me: that "noir", "lonely", and "complaint" were both associated with positive reviews, and that "redeeming" is associated with the negative category. The last makes sense because it would like be in a "a one redeeming quality of this movie is...". Similarly, "complaint" being associated with positive reviews could be logical in the sense that people would write "my only complaint...". Noir and lonely being associated with positive reviews is strange to me, however, and should be investigated further.

Identify the top 10 features with the maximum positive weights for POSITIVE category.

['wonderfully', 'noir', 'delightful', 'excellent,', 'lonely', 'perfect.', 'perfect,', 'complaint', '8/10', 'can.']

Identify the top 10 features with the maximum negative weights for POSITIVE category.

['redeeming', 'insult', 'disappointment', 'unfunny', 'horrible.', 'dull,', 'wasting', 'garbage.', 'unconvincing', 'pathetic.']

Identify the top 10 features with the maximum positive weights for NEGATIVE category.

['redeeming', 'insult', 'disappointment', 'unfunny', 'horrible.', 'dull,', 'wasting', 'garbage.', 'unconvincing', 'pathetic.']

Identify the top 10 features with the maximum negative weights for NEGATIVE category.

['wonderfully', 'noir', 'delightful', 'excellent,', 'lonely', 'perfect.', 'perfect,', 'complaint', '8/10', 'can.']

## Questions

While I understand overall what the model is doing and what each piece is of the code is achieving, I'm still confused about what the `loss.backward` and `optimizer.step` do specifically. I understand that the softmax function calculates predictions given that the model is a logistic regression, and I understand that the loss function is calculating how incorrect and/or how far off those calculations were. Does the `loss.backward` step then go back up the layers/nodes, correcting previously assigned weights as it goes back to the top? Then the `optimizer.step` does another pass to correct that?

Related to code, I noticed that some tutorials would load the same model and tune it specifically, but I couldn't understand exactly how that was happening to do it. Do you have any resources I could look at for future assignments?

Lastly, my computer has an Nvidia card in it. Are we allowed to push the calculations off to that for these assignments to make them run faster?