Problem 1

* 1. From the hinge loss definition:

We can calculate the gradient piecewise:

We also want to run sgd, updating the weights according to:

The problem also required that we use

(-1) pretty bad

|  |  |  |  |
| --- | --- | --- | --- |
| y | **w**old |  | **Wnew** |
| -1 | [0, 0, 0, 0, 0, 0] | [1, 0, 1, 0, 0, 0] | [-.5, 0, -.5, 0, 0, 0] |

(+1) good plot

|  |  |  |  |
| --- | --- | --- | --- |
| y | **w**old |  | **Wnew** |
| +1 | [-.5, 0, -.5, 0, 0, 0] | [0, 1, 0, 1, 0, 0] | [-.5, .5, -.5, .5, 0, 0] |

(-1) not good

|  |  |  |  |
| --- | --- | --- | --- |
| y | **w**old |  | **Wnew** |
| -1 | [-.5, .5, -.5, .5, 0, 0] | [0, 1, 0, 0, 1, 0] | [-.5, 0, -.5, .5, -.5, 0] |

(+1) pretty scenery

|  |  |  |  |
| --- | --- | --- | --- |
| y | **w**old |  | **Wnew** |
| 1 | [-.5, 0, -.5, .5, -.5, 0] | [1, 0, 0, 0, 0, 1] | [0, 0, -.5, .5, -.5, .5] |

Final weight vector: [0, 0, -.5, .5, -.5, .5]

* 1. A = {not: w1, bad: w2, good: w3}

Dataset = {good: w3 > 0, not good: w1 + w3 < 0, bad: w2 < 0, not bad: w1 + w2 > 0}

Based on this data set, no feature can get zero error because we have a contradiction, based on w1 + w2 > 0 and w1 + w3 < 0. Because we have already defined w3 > 0 and w2 < 0, we run into issues with w1 somehow being > 0 and < 0. Adding the additional feature not bad: w4 to A would mean that we could remove and avoid this contradiction. The new dataset would be {good: w3 > 0, not good: w1 + w3 < 0, bad: w2 < 0, not bad: w4 > 1}, for example.

Problem 2

1. d
2. For the prediction below, I believe the classifier would need more information for bite and humor, because most of the other words are less common or unique to each individual review.

Graphical user interface, text

Description automatically generated

1. Graphical user interface, text

   Description automatically generatedFor the prediction below, I believe the classifier would need more information on words like ever, epic, and be, whose weights are very high and would need more data to make a more accurate prediction.
2. Graphical user interface, text

   Description automatically generatedThis classifier prediction below would more information on words like film and lacks because the weights are very high and need more data to be properly weighed.
3. This classifier would need more information on ride, modern, through, which, and to, which are all words that are weighed very high and may be throwing off the prediction.

Text

Description automatically generated

1. This classifier would benefit from having more information on words like films and sweet, as they are weighed very high and may be throwing off the prediction.

Graphical user interface, text

Description automatically generated

1. 2f

I would explain the similar error to the way both feature extractors work. Using stochastic gradient descent is very similar (in learning) to a simpler method, which involves just mapping each string of n characters to the number of times it occurs. If I were to try to construct a review, I would try to say something along the lines of, “The movie did a pretty good job of managing actors and a pretty good job with the plot,” using the logic that n-gram classification would handle duplicates better than word features.

Problem 3

1. Adding the ‘entity is’ feature string before every feature seems to have drastically minimized the error we received from the data. The new feature tries to find features in the data to see if they are a valid entity, and if we have predicted them correctly. The validation example that is correct classified correct entities like the senate and Margaret McCullough, which without entity is were incorrect. The learned features helped from the evidence that the training and validation error rate were significantly lower, and also upon visual inspection of the error analysis, we can tell that it is doing better at correctly recognizing entities.
2. The new features added were the “left is” and the “right is”. An example being instead of just Eduardo Romero, we now read , Eduardo Romero (. I believe it was now able to correctly classify Amman and Andre Caboche, but it also now incorrectly predicted “. Margaret McCullough ,”. The learning features helped because the validation error rate was now lowered from the before the two features were added.
3. The newest feature added this time around was checking through the entities inside the entity produced from 3a. For example, instead of saying “entity contains: Eduardo Romero” we now can check “entity contains: Eduardo”. Validation examples like Felix Mantilla are now correct because of the new feature, along with the Kurdistan Workers Party. They are both now correct apparently primarily because of the new features. Besides having new correct validation examples, the training error rate was still low, and then the test error continued to improve with the addition of the features in 3c.
4. The new feature breaks the word entity in half, into a prefix and suffix. A good example for this would be in “entity is Sarah Pitkowski”. We now also have the entity containing: “Sarah, suffix arah, Pitkowski, suffix wski, and prefix Pitk” This adds more variation to the data, and in turn, more features to learn from. While I couldn’t find any specific examples, I can say that there were definitely features that were now correctly predicted, because doing a basic “ctrl-f” to find the number of times WRONG occurs, we can see that part d had less occurrences of incorrect values than part c. Additionally, we can say that the learned features helped because we saw an improvement in validation test error again.