

1. Leo Zhuang llzhuang@usc.edu
2. I chose to use a classification task mainly because I wanted more practice with neural networks. More importantly, I felt that the valence of a tweet was better captured by a discrete system. A really positive tweet isn't very different from a moderately positive tweet in my eyes. I mainly wanted to find whether it was positive or not.
3. I definitely wanted to use word embeddings to encode the tweets. Because it was already and quickly available to me, I used the Glove embedding.
4. I only trained the model on the tweet text contents, and my target was valence.
5. I used an RNN, with an embedding layer, and two LSTM layers. One had input size of 34 by 50 (34 word tweets, 50 size for embedding), and the other had an input size of 64. Finally, there was the output layer of dimension 1. The two hidden layers had relu activation, while the output layer was sigmoid. I was basing this project mainly off the RNN lesson that we did in curriculum, in which I used an LSTM layer with relu then a LSTM with tanh. I found that this didn't really work for this project. I noticed that the model was learning very slowly, so I tried using tanh, since it has more effect for smaller numbers. However, this didn't work. In the end, I decided on relu for both through trial and error.
6. I used binary cross entropy for my loss, since I ended up making my valence binary, either positive or negative, since all the data was either valence 0 or 4. Since my results were not ideal, I tried different optimizers. I didn't notice much results, and went with ADAM after combing through some resources on the internet.
7. I took 0.3% of all the tweets since there were a lot of tweets, and then made 33% of them into the test set, with the remaining as a training set.
8. I just used a simple loss and accuracy to measure how well the model did. Unfortunately, I only reached an accuracy of 68%, probably due to something in my pre processing.