**IMDB Code manipulation:**

Steven Kalinoff  
2/18/2024

Adv. Machine Learning 64061  
  
**Control**  
The Control Data was used as a baseline to show how well the process learned over 20 Epochs. The data itself was a score of an item on IMDB and a review attached to that score. The goal of the model is to build a relationship between the score and words used to (10,000) most common in this case. Simply put these reviews were picked as 5/5 good or 1/5 bad and we are going to binary train and test our data on whether a review is good or bad based on the words the author has chosen.

We see an exponential decrease in in training loss epoch over epoch but the validation loss positively over the 3rd plus epochs. Our model is trying to overfit the data at this point. We can confirm this with our accuracy graph that the training data is getting more refined, but we stop seeing validation data improvements after the 3rd epoch and start to level off.

A graph with blue lines

Description automatically generatedA graph with blue dots

Description automatically generated

We are then asked to modify this neural network to explore and document what happens when we tweak certain parameters.

**Adding a hidden layer**

By added a hidden layer we see the model trains data almost the same however our loss in our validation shoots up rather fast at the 3rd epoch. We also see that the accuracy on validation data levels off at this point. I hypothesize that the extra hidden layer is making too many connections to the data and is overfitting.

A graph of a training loss

Description automatically generatedA graph with blue dots

Description automatically generated

**Adjusting Hidden Units**

When adjusting the number of hidden units, when adjusting the hidden units, we are effectively increate the dimensionally of the model allowing it to create more complex connections. I chose to increase the number of hidden units to 32 from 16. An interesting patten in later epochs occurred in the training data. Both the loss and accuracy started to oscillate. The training loss is the worse we’ve seen yet approaching 60% and validation accuracy doesn’t improve.

A graph of a line

Description automatically generatedA graph with blue dots

Description automatically generated

**MSE Loss**

Changing the loss function from binary\_crossentropy to MSE. This model greatly reduces the loss function overall. This model also gets worse when more epochs are applied and appears to start trending downward in later validation accuracy tests.

A graph with blue dots

Description automatically generatedA graph with blue dots

Description automatically generated

**Tanh**

Changing the activation element from relu to an older method called Tanh has a very similar validation loss spike in higher epochs as the control model we have used. In the training accuracy we can see the inverse result in the accuracy around epoch 16. This method also appears to be trending downward in terms of accuracy in validation.

A graph with blue dots

Description automatically generatedA graph with blue dots

Description automatically generated

**Dropout**

Our final task was to add a technique we studied in class to attempt to improve model performance on the validation task. I chose dropout, this is a function that randomly “drops” connections in training to kill those formed links and forcing the next layer to adapt to this. In theory this method allows the neurons to learn the hidden features better. During this trial the training loss dropped sharply in the first few epochs but then erratically rose in later. On the accuracy side we see a sharp increase in validation accuracy after the first few runs, and it remains relatively stable after this showing that the method did increase validation accuracy.

A graph with blue dots

Description automatically generatedA graph with blue dots

Description automatically generated

**Conclusion**

After looking at all 6 iterations of the experiment I have concluded that various rules can be applied to this experiment to achieve the highest possible likelihood of validation accuracy.

* The number of Epochs should be constrained to correlate with the local minima between 1-x. In this case 1-5 should cover all examples.
* Use a MSE loss function will decrease validation loss while proportionally increasing validation accuracy.
* Combining these above with a dropout function before each of the first two hidden layers will have the model with the smallest chance of overfitting the data.