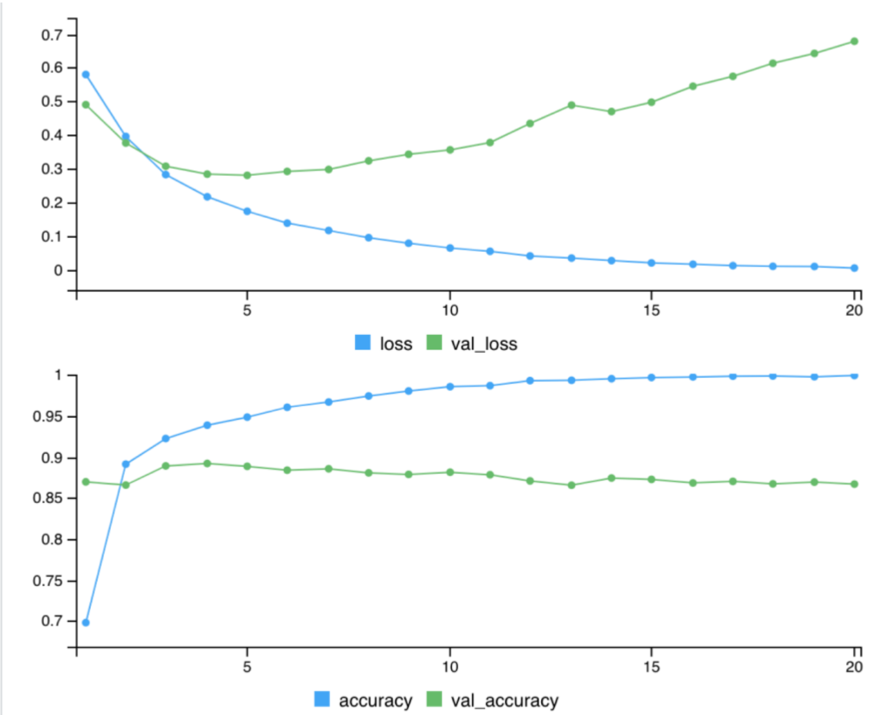
Julia Thacker
Assignment 1 – Advanced Machine Learning

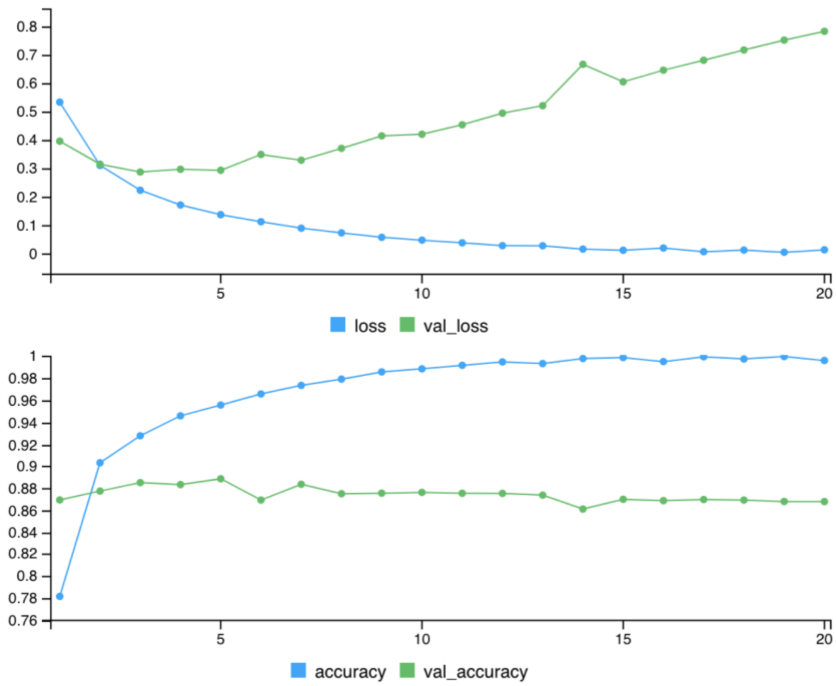
I started by visualizing the original model from the class example, which can be seen in the graph below. You can see that the validation loss steadily increases while the training loss decreases. The validation accuracy was also notably less that the training accuracy.

Original Model Example test:



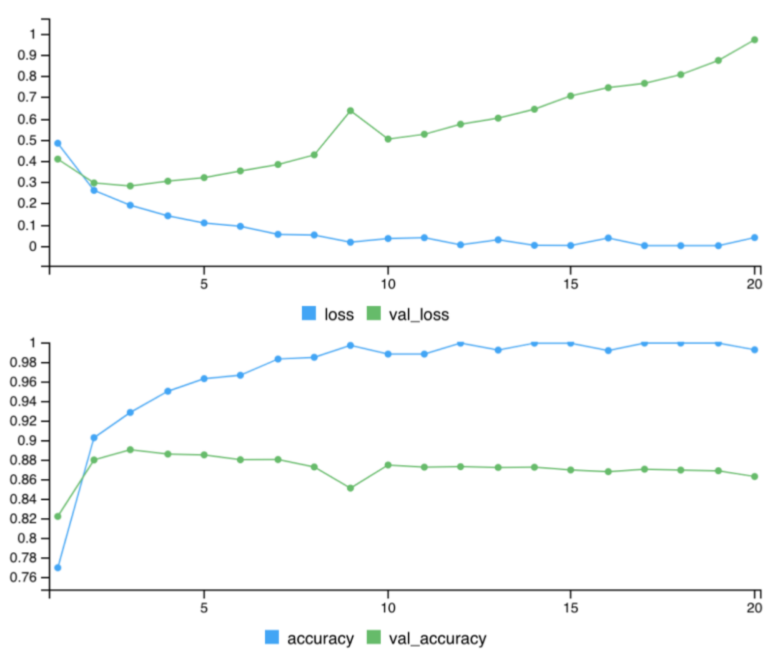
My first test was to create a model that had three layers instead of the original two, and the visualization of that model can be seen below. The validation loss is steadily increasing in this model as well and appears to be on a slightly steeper incline. The validation accuracy also did not increase from the original model. Overall, this attempt did not improve the original model.

Model 2 – 3 layers:



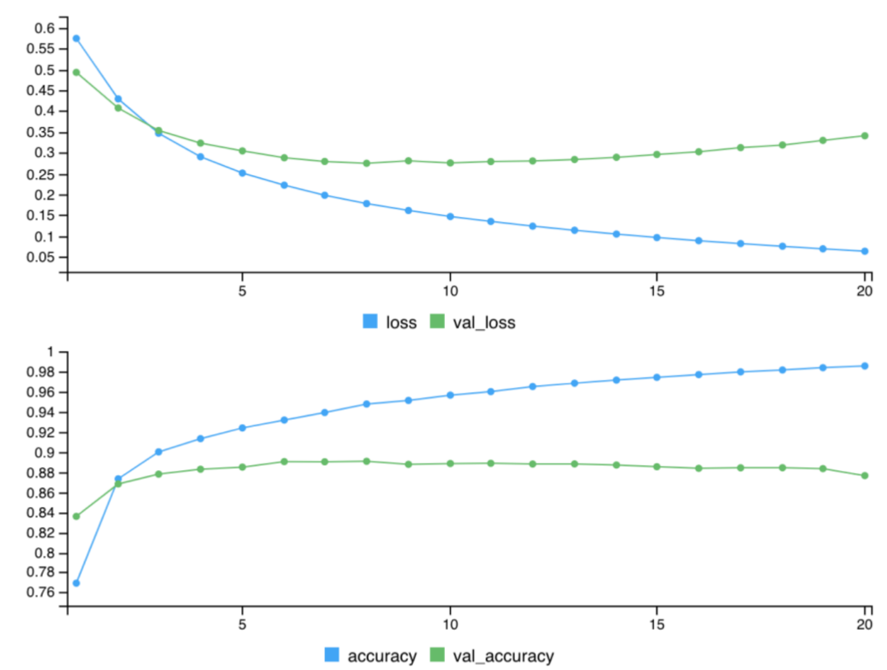
Next, I created another model where I changed the number of units from 16 to 64 units. This model also was not an improvement from the original. The validation loss has increased and the accuracy has not.

Model 3 – 64 hidden units:



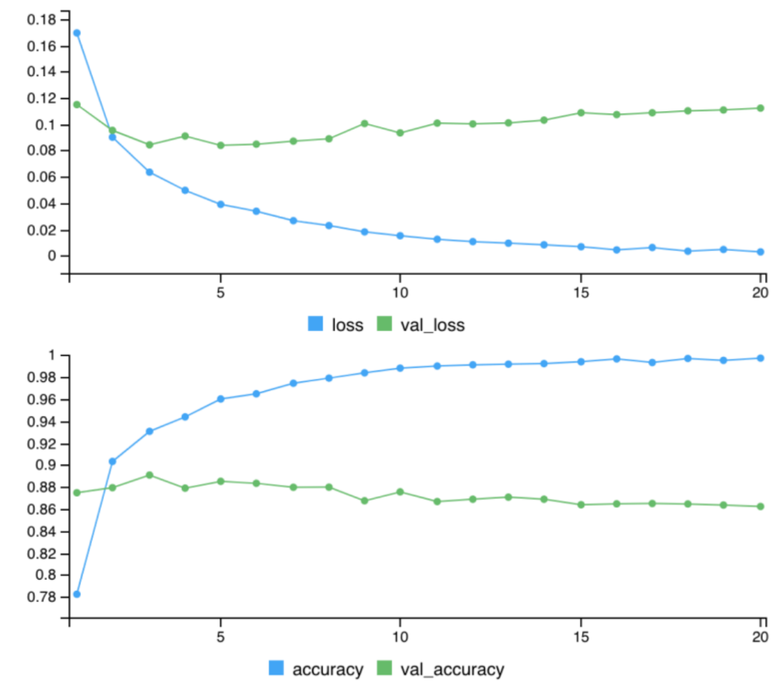
I then adjusted the hidden units again to see the impact of a significantly smaller number of hidden units after testing a large number in the prior model. Model 3b below has 2 hidden units instead of the original 16. This model had the most improvement of the tests I have done so far. The validation loss is not increasing as significantly as the previous two models, and the validation accuracy remains steady with each epoch.

Model 3b – 2 hidden units:

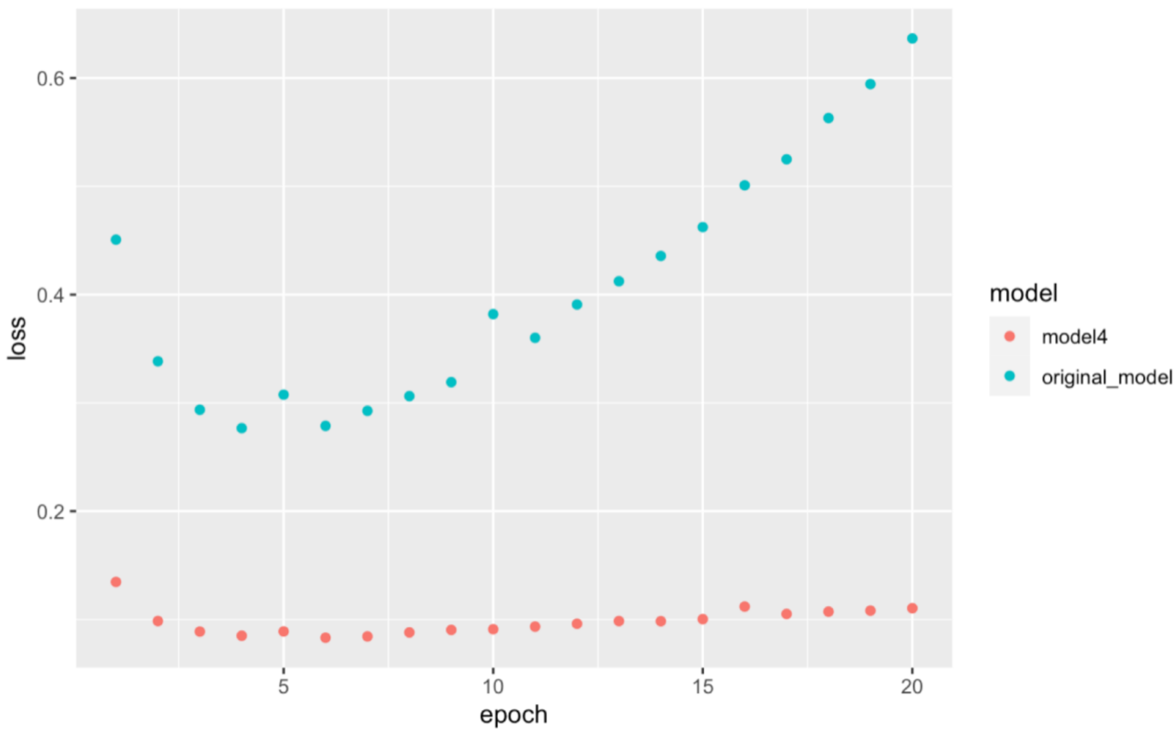


For the next model, I returned the units to the original 16 and used the mse loss function instead of binary_crossentropy. This model then became the best performing so far with significantly lower validation loss. The validation accuracy is on a slight decrease but remains relatively high.

Model 4- mse loss function:

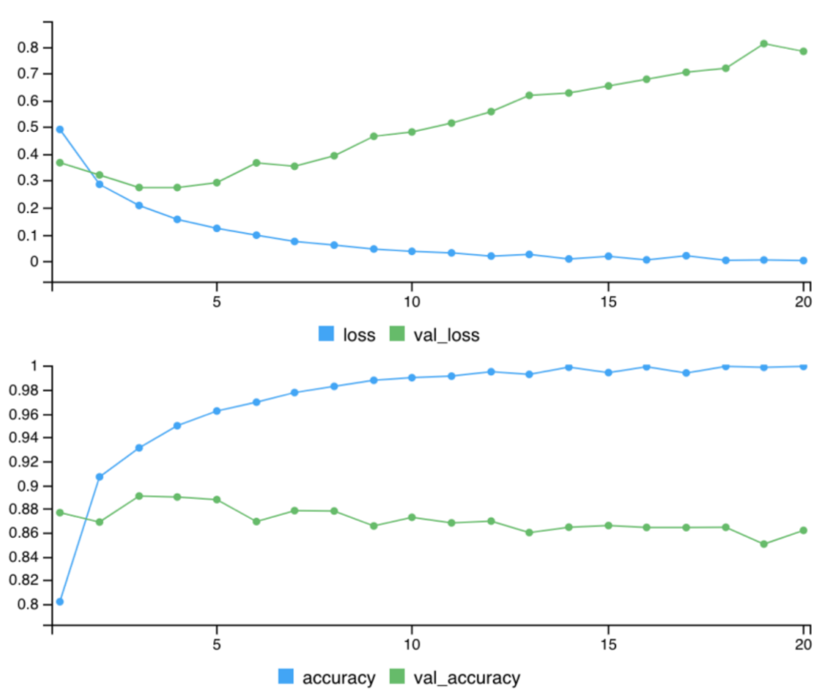


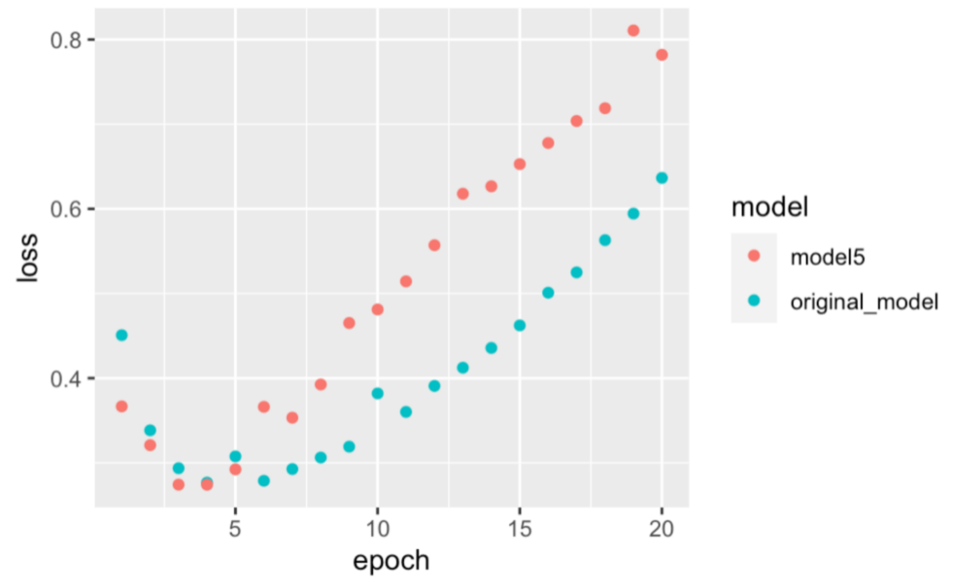
I also made an additional visualization that shows the improvement from the original model by using the mse loss function. The loss of the new model is almost nonexistent in comparison to the original model.



The next test that I performed was using tanh activation instead of relu. This model did not produce better results. As seen in the visuals below, the validation loss is steadily increasing, and the validation accuracy is decreasing. We can also see that compared to the original model, this new model is more susceptible to overfitting.

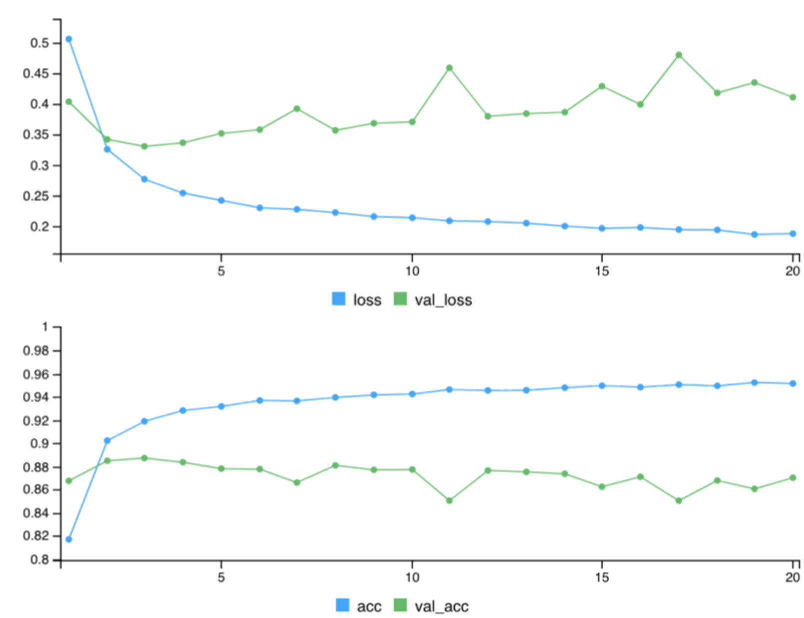
Model 5 – tanh activation:

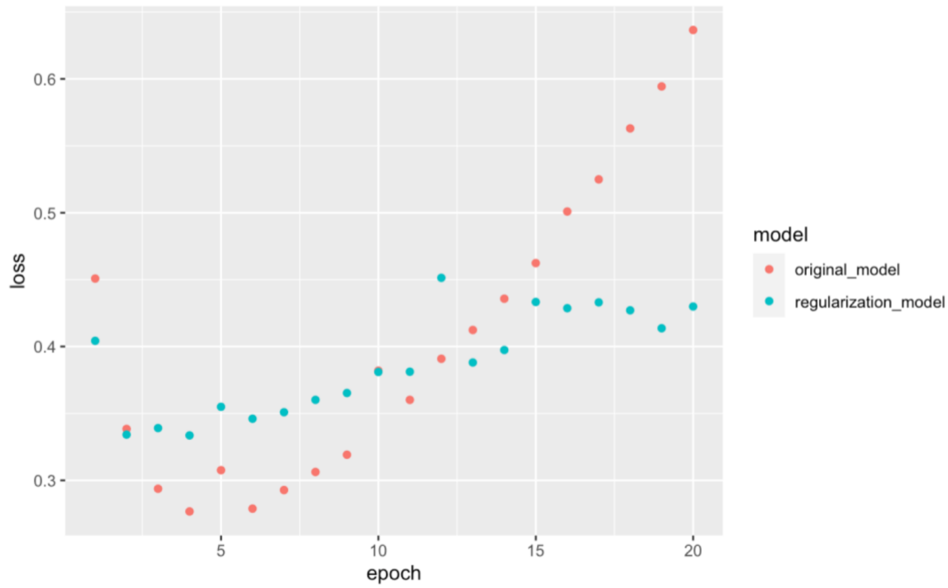




In my final attempts to produce a model that would perform better on validation, I decided to use the regularization technique. I started with a weight of .001.

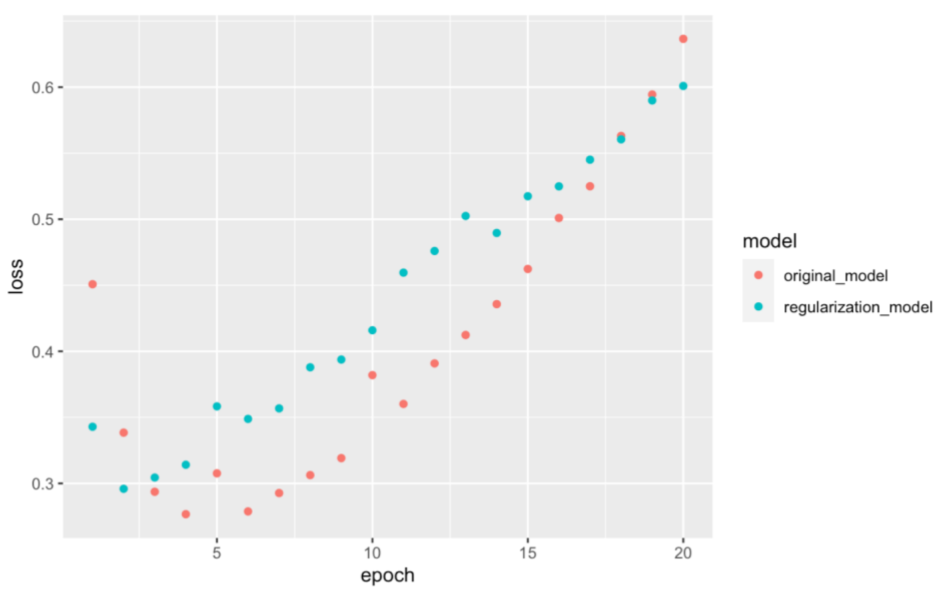
Regularization model, 0.001 weight:





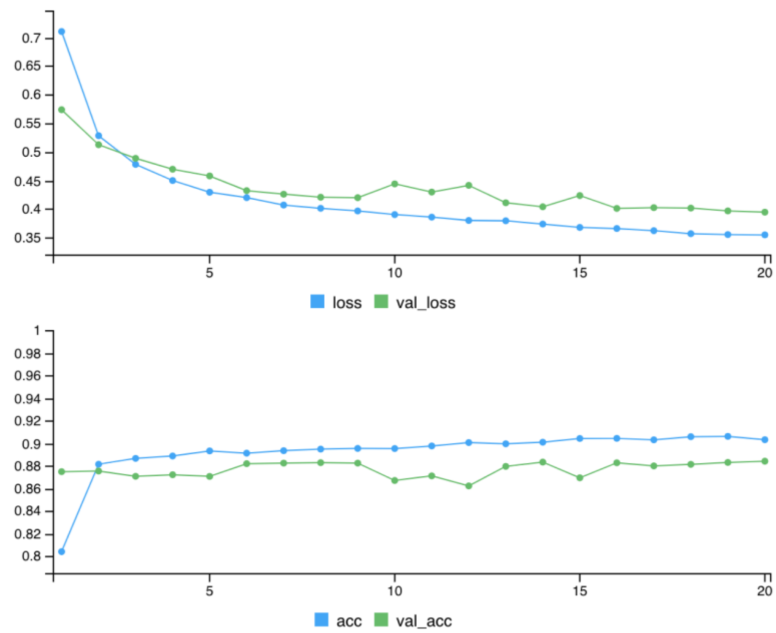
I then decreased the weight to only .0001 to see what that would look like. This performed worse than the previous attempt, so I decided to try to increase the weight slightly.

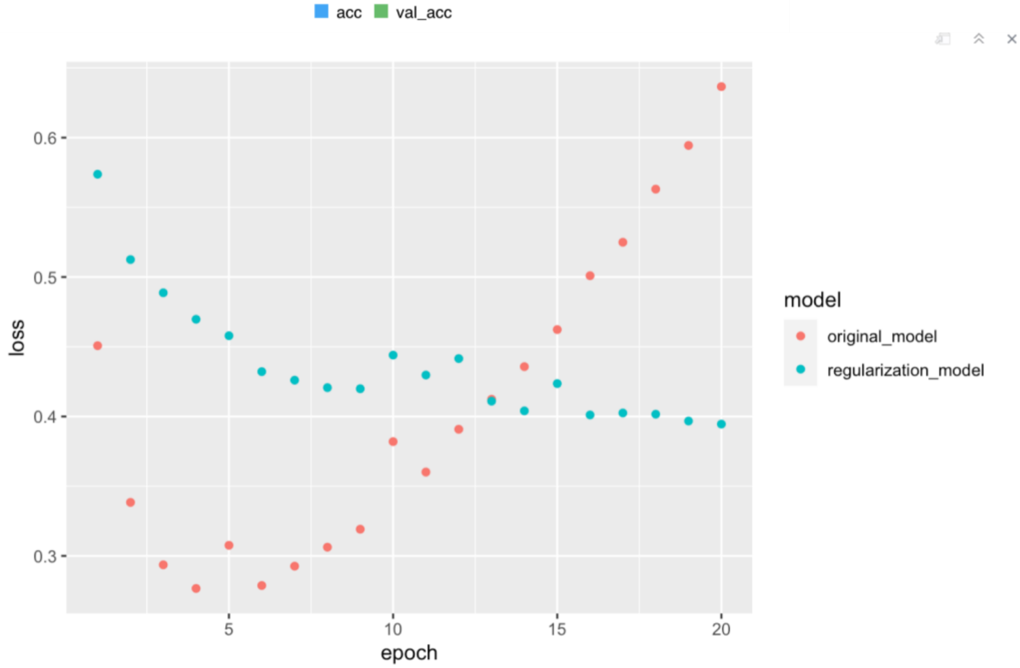
Regularization model, 0.0001 weight:



I increased the weight to .01 to see if that would help the model perform better. In the visual below, you can see that the validation loss and accuracy is much more in line with the training loss and accuracy. The loss is significantly lower than previous models.

Regularization model, 0.01 weight:





My final decision was to keep the .01 weight and to also use the mse loss function as that seemed to work well in my previous experiment with it. Both the training and validation loss are siginificantly lower than any other attempt. The two are also very close together except for a few minor stray points in validation. The accuracy for both training and validation also remain in line and do not drop off. The second visual shows that compared to the original model, this final model is less noisy and much more resistant to overfitting.

Regularization model, 0.01 weight and mse loss function: