In order to classify movie reviews on IMDB as positive or negative, a model is being developed. The model will use previously labeled (positive/negative) reviews with the goal of being able to classify unlabeled reviews as accurately as possible. The dataset is comprised of 50,000 IMDB reviews that tend to be more polarized in a positive or negative direction. The model will be assembled using Python, through Keras and TensorFlow.

The model will need to be tuned accordingly to optimize performance, as some hyperparameters are better suited for different types of models. Various combinations of hidden layers and hidden units were explored, as were different activation functions such as *relu* and *tanh*. Different loss functions (specifically mean squared error/MSE and binary cross entropy), and regularization were explored as well.

The base model included the relu activation function, 2 hidden layers, 16 hidden units, and a binary cross entropy loss function. Different versions of the model were tested, with the validation accuracy (should be closer to 100%, or 1) and validation loss (should be closer to 0) being measured. The results are below:

	Tuning	Validation Accuracy	Validation Loss
0	Base Model	0.870000	0.560000
1	1 Hidden Layer	**0.880000**	0.380000
2	3 Hidden Layers	0.870000	0.630000
3	8 Hidden Units	0.870000	0.470000
4	32 Hidden Units	0.870000	0.600000
5	MSE	**0.880000**	**0.100000**
6	Tanh	0.870000	0.740000
7	Regularization	0.870000	0.430000

The optimal results are highlighted accordingly. Adjusting the hyperparameters still resulted in relatively consistent validation accuracy. However, the validation loss varied widely throughout different tunings. While the validation accuracy was relatively consistent throughout, both the "1 Hidden Layer" and MSE tunings produced the highest accuracy. For validation loss, MSE was decisively the best performing (lowest) tuning. As a whole, it appears that the simpler tunings performed better than the more complex tunings, suggesting a simpler model with fewer hidden layers/units and incorporating MSE may be the best option. Using a simpler model can also use less resources.