Isaac Simoneau | Benjamin Gervais | Andrew Bastress
Final Results

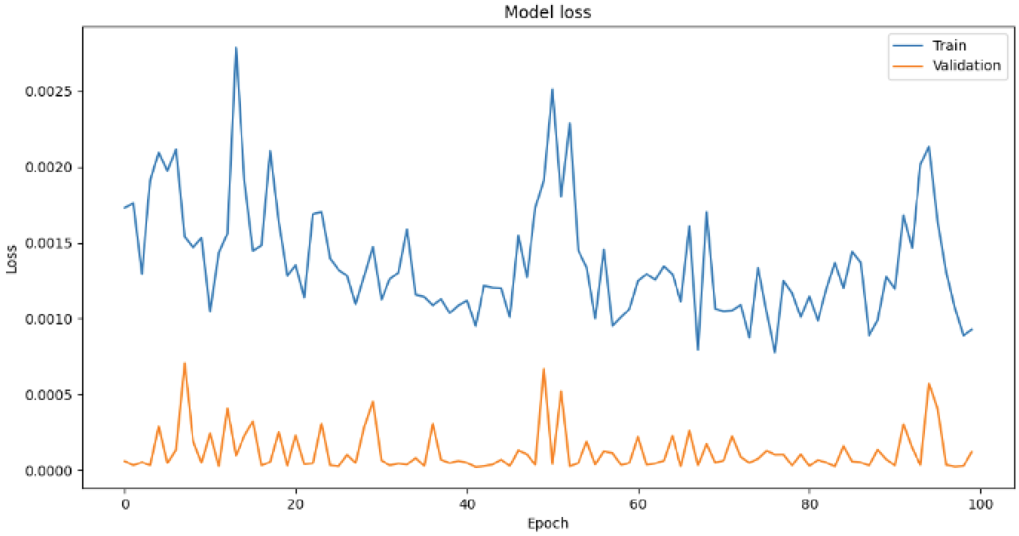
Final Training Results

The preliminary results of our LSTM model were positive, but were inconsistent as we hadn't yet landed on the optimal hyperparameters for our model. The MSE ranged from 20,000 to 2,000, and it'd vary greatly even with the same hyperparameters (batch size and epochs being the only ones thoroughly tested).

In comparison to our preliminary results, we've reduced the MSE of our results to around 400. We experimented with various hyperparameter changes, the following leading to the greatest improvement:

- Dropout rates - our three hidden layers all had dropout rates of 0.2 previously. By changing them to 0.8, 0.1, and 0.1 respectively, we improved the models accuracy by preventing overfitting.
- Hidden layer units - the number of units in the hidden layers were increased from 50 to 100 each, allowing the model to retain more complicated and nuanced information.
- Validation split - while optimizing, we incorporated a validation split to ensure the model could be generalized and prevent overfitting. As seen in the graph below, the model performed exceptionally well on the validation set.

Additionally, we experimented with the learning rate and various Keras optimizers in the hopes of improving performance. We tried AdamW and AdamDelta optimizers, but stuck with Adam using the default learning rate of 0.001 in the end.



Final Demonstration Proposal

Going forward, the goal would be to set up a web app allowing users to input any ticker symbol, triggering an API request which would run our model on data from that company’s closing prices, ultimately outputting a predicted price. However, we’re going to have to start smaller.

We plan to set up a simple but effective web page using the Flask/PyTorch approach detailed in the MAIS 2021 tutorial. Our model was trained on the daily closing price of Google, so if we can integrate the Alpha Vantage API, we could pull the latest data at the click of a button and output the prediction for the current date. Otherwise we’d have to run on older data, or manually fetch the data ahead of time. Either way, we’ll be able to demonstrate the capabilities of our model by displaying a predicted price when prompted by the user.