**Stock Price Prediction – Costco**

**Deep Learning Model: GRU 🡪 LSTM🡪 Fine Tuning**

**TABLE OF CONTENTS**

1.0 Introduction 3

1.1 Problem Statement 3

1.2 Data Description 3

2.0 Methodology 5

2.1 Data Preprocessing 5

2.2 Data Visualization 6

2.3 First Trial: GRU Model with multiple inputs 6

2.3.1 Model A-1: GRU with 6 Time-Series as inputs 7

2.3.2 Model A: Performance evaluation & Hyper-parameter Optimization 7

2.3.3 Model B: LSTM with 6 Time-Series as inputs 9

2.3.4 Model B: Performance Evaluation & Hyper-parameter Optimization 9

2.3.5 Model C: LSTM with as single input 12

2.3.6 Model C: Performance evaluation & Hyper-parameter Optimization 12

3.0 CONCLUSIONS 14

4.0 REFERENCES 15

# Introduction

In this project, we aim to explore the potential of applying deep learning model in stock price prediction. Specifically, we will focus on predicting the daily closing price of COSTCO, a consumer staples stock, using historical stock market data, foreign exchange (FX) rates, interest rates, credit spread, and relevant economic indicators.

There are two assumptions in this project:

Hypothesis 1: The price movement of consumer staples tends to be more stable compared to other sectors, and thus might be easier to build a model for prediction.

Hypothesis 2: Macroeconomic indicators, such as interest rates, can have a significant impact on the stock prices of COSTCO.

By examining the relationship between these factors and the stock prices of COSTCO, we aim to provide insights into the market and help investors make informed decisions. Our results will be evaluated based on the accuracy of our prediction model and the validity of our hypotheses.

## Problem Statement

To predict closing price of COSTCO, a consumer staples stock, with historical price of the stock market, FX, interest rate data and relevant economic indicators

## Data Description

|  |  |
| --- | --- |
| **Data period** | **From year 2000 - 2022**  **(Around 5700 Trading Days)** |
| **Data Split** | **Training: 2000 - 2015**  **Validation: 2016 -2019**  **Testing: 2020- 2022** |

Table 1：Data Period & Data Split

|  |  |  |  |
| --- | --- | --- | --- |
| **Features by Category** | **Dataset** | **Data Source** | **Consideration** |
| **Stock Price (Closing)** | **Stock price of the same sector**  Costco (COST)  Coca-cola (KO),  P&G (PG)  Walmart (WMT), Nestle (NSRGY) | Google Finance | Stocks from the same sector should have similar movement.  Earnings results from other company may impact target value |
| **Interest Rate** | **US Treasury Yield**  6-month, 2-year  10-year | US Treasury website | Higher interest rate would result in lower stock valuation |
| **Foreign exchange** | **US Dollar Index (DXY)** | Yahoo Finance | Higher DXY would result in lower stock valuation |
| **Credit spread** | **Credit Spread**  US Corporate Grade OAS spread Index | ICE | Higher credit spread means higher credit corporate default risk and therefore higher |
| **Economic Indicators** | **Inflation measure**  CPI  **Labour market**  Unemployment rate | US Federal | CPI/ unemployment would affect decision on monetary policy and therefore the stock market. |
| **TARGET VALUE** | **Daily closing price** | Google Finance | N/A |

Table 2: Data Description

Since the initial hypothesis was to add more feature variables and allow the model to choose which features it should weigh more, we included the stock prices data with 6 consumer staples companies, (around 5700 trading days of data), and some economic indicators such as Inflation measure (Consumer price index) and Unemployment rate.

For this project, the dataset was split into Training dataset: Year 2000 – 2015, Validation dataset: Year 2016 –2019, and Testing dataset: Year 2020 – 2022. All of data collected are from valid and accountable sources namely, US Federal Reserve, Yahoo Finance, Google Finance, etc.

# Methodology

## Data Preprocessing

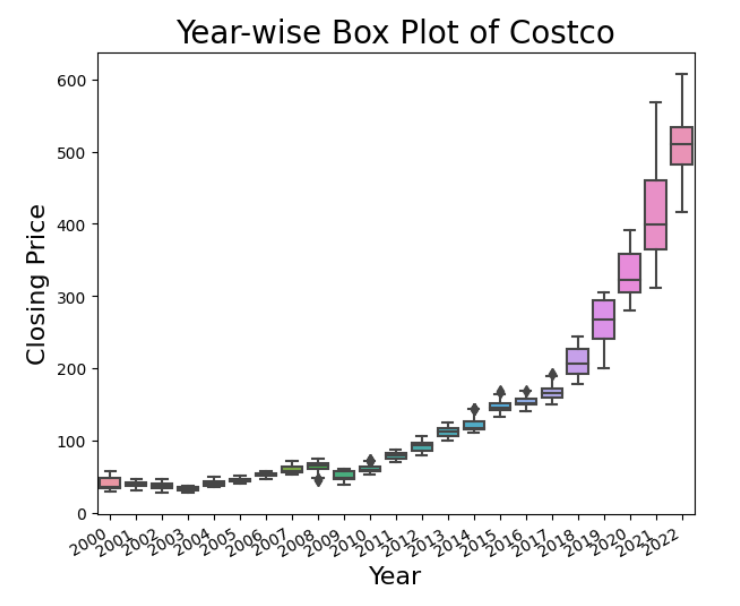
|  |  |
| --- | --- |
| Costco | 1 |
| Coca-cola | 1 |
| Nestle | 60 |
| Interest rate | 44 |
| GDP | 63 |
| US Dollar index | 114 |
| Credit Spread | 2 |

Table 4: Missing Data Summary

There were some data missing in stock data file and economic indicators data file as shown in the above table. Backfill method is applied to fill the missing value.

* + Data format is aligned with yyyy-mm-dd.
  + Concatenate desired stocks and economic indicators
  + Replace null values
  + Window size = 3
  + MinMaxScalar is applied to scale the numerical data to same range between 0 and 1
  + Data Split:
    - Training data: Year 2000-2015
    - Validation data: Year 2016-2019
    - Testing data: Year 2020-2022

## Data Visualization

  
Figure 1: Box plot of Costco Closing Price from 2000 to 2022

The trend of Costco closing price has been increased from 2000 to 2022. It showed that Costco is a good choice for investment.

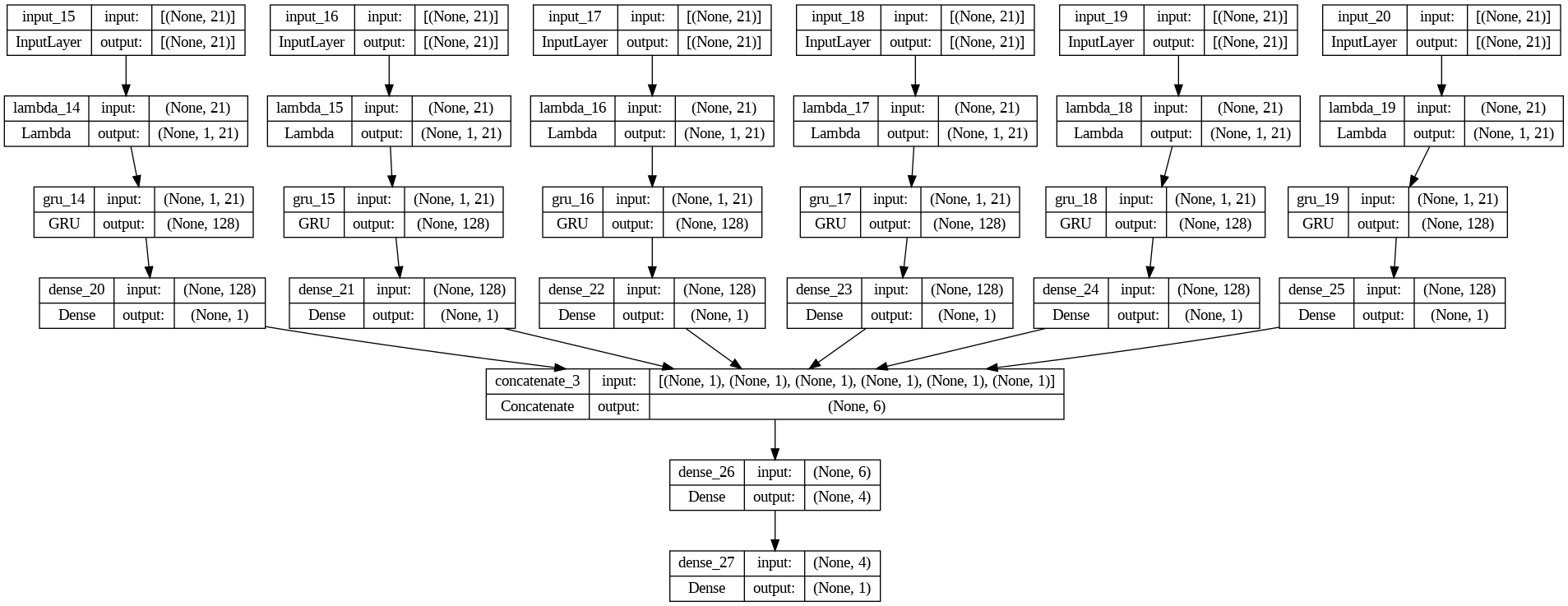
## First Trial: GRU Model with multiple inputs

For the first trial, we take all the factors we prepared into consideration and build a multiple input GRU model which can process and analyze multiple sequences of inputs simultaneously. This can be useful in cases where multiple related sequences need to be considered to make a prediction, as the model can learn to weigh the importance of each input sequence and integrate the information in a meaningful way. This could improve accuracy compared to a single input model, especially when the relationships between the sequences are complex.

In a big picture, we would build a multiple-branches GRU Model. The reason for using different branches in a multiple input GRU model is to allow each branch to specialize in processing a specific type of input sequence. Each branch can learn to extract relevant features and make predictions based on its own specialized input sequence, and the final output can be obtained by combining the results from each branch. This allows the model to handle input sequences of different modalities or forms and make predictions based on a more complete representation of the input data.

### Model A-1: GRU with 6 Time-Series as inputs

GRU with 6 time-series as inputs, e.g., stock data for Walmart, Costco, P&G, Pepsi, NSRGY and KO was used to predict the future price of Walmart:



*Figure 2: Model A-1 GRU with 6 Time-Series inputs*

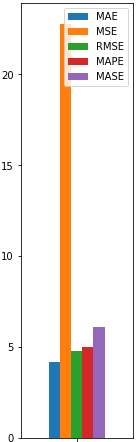
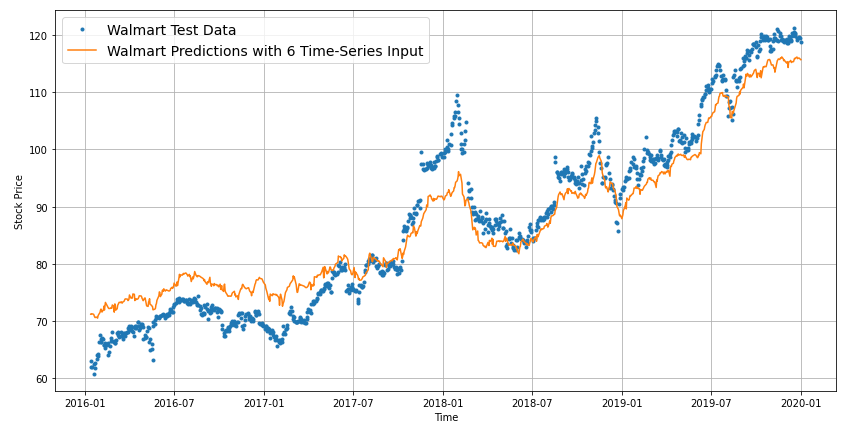
Each branch consists of 1 GRU layer with 128 neurons and one Dense layer with a single neuron. The GRU layer uses a Rectified Linear Unit (“ReLU”) activation while the Dense layer has a linear activation.

After concatenation, there is one Dense layer with 4 neurons in series with an output layer having a single neuron. Both Dense layers use the linear activation. It was found that linear activation for the output layer worked best for our case.

This architecture was our starting point. The intention was to improve on our model with hyper-parameter tuning, but this plan was abandoned once it was clear that our original model does not produce accurate results.

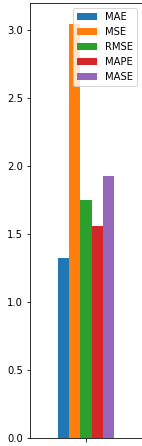
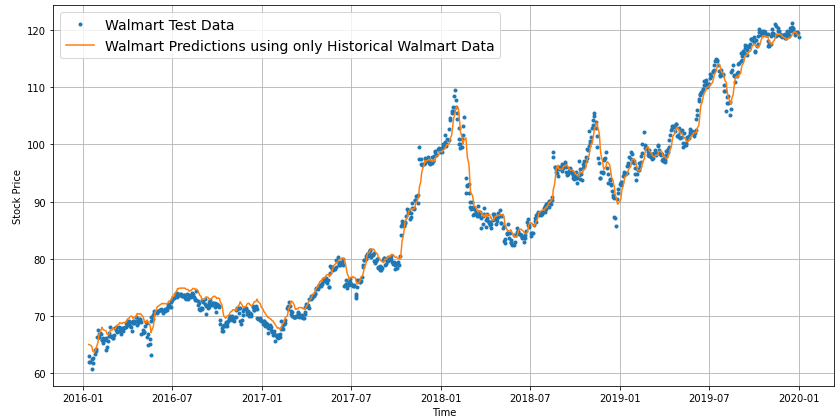
### Model A: Performance evaluation & Hyper-parameter Optimization

Let’s look at the initial result after training Model A-1 with historical data from all 6 stocks (Walmart, Costco, P&G, Pepsi, NSRGY and KO) from the same sectors to predict future price of Walmart.



*Figure 3: Walmart Prediction Using Original GRU Model A-1*

We compare the result with that of model A-2 that uses only historical Walmart data to predict future stock price:



*Figure 4: Walmart Prediction with Model A-2 Using Only Walmart Historical Data*

Model A-2 result is much more accurate than that of our original Model A-1. The results show us that having more features does not necessarily make our predictions better. We also need to know if the data is relevant to our target.

Further Improvement with hyper-parameter Tuning:

We continued to do engineering work to improve our model by varying the hyper-parameters. Specifically, by varying the learning rate and the number of epochs, we were able to reduce the variations in the loss function during training.

|  |  |
| --- | --- |
| *Figure 5: Training loss against epoch (lr = 0.001)* | *Figure 6: Training loss against epoch (lr = 0.01)* |

### Model B: LSTM with 6 Time-Series as inputs

Diagram, engineering drawing, schematic

Description automatically generated

*Figure 7:* *Model B LSTM with 6 Time-Series inputs*

The model takes six different inputs and processes each input through two LSTM layers. The first layer has units\_l1 units and is set to return sequences, while the second layer has units\_l2 units. The outputs of the second LSTM layer from all inputs are concatenated and passed through two dense layers with 128 and 32 units, respectively. The final dense layer has 1 unit and uses the Rectified Linear Unit (“ReLU”) activation function.

The input of the model is 3 days prices of 3 stocks with 3 indexes and the output of the model is the next days close price. Due to the additional gates and memory cells in the LSTM architecture, which allow for more nuanced processing of the input data, so that we decided to test if it improves the accuracy and performance. Also, with hyper-parameter tuning, we plan to optimize the model to do the further improvement.

### Model B: Performance Evaluation & Hyper-parameter Optimization

The hyper-parameters that we are testing.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Hyper-parameters |  | | | |
| Optimizer | Nadam | Adam | SGD | RMSprop |
| Batch Size | 1 | 8 | 16 | 64 |
| Learning Rate | 0.001 | 0.0005 |  |  |

*Table 5: Hyper-parameters*

During the presentation, we did not show the whole picture of result due to the complexity.

A picture containing chart

Description automatically generated

*Figure 8: Prediction using different set of combinations of hyper-parameters, comparing to the testing data.*

Most of the predictions do not follow the trends of the actual prices. The possible reason is that those models exist overfitting and are unable to learn properly. Therefore, the next step of the project will be to reduce the parameters in the model to prevent overfitting. In case, those models can catch moving features.

Chart

Description automatically generated with low confidence

*Figure 9: Actual versus predicted movement using different set of combinations of hyper-parameters, comparing to the testing data.*

The values momentum graph showed the momentum of the predictions are following the actual values before 2020, but the values become separated. The possible reason is that the features of the stock price before 2020 and after 2020 are different. Plus, overfitting causes the predictions after 2020 to be wrong.

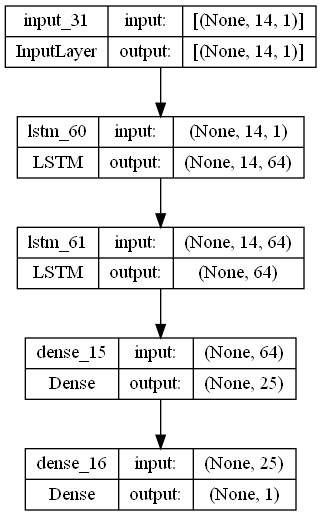
Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated*Figure 10: Training Loss and Validation Loss against epochs*

After having a big picture, we pick 1 suitable model (i.e. Adam for the optimizer, and batch size of 32) for further study, and evaluate the validation loss. The loss is reduced in an abnormal quick rate in training, which the loss is fluctuating in validation stage. It is a further reason that the model is learning improperly. Therefore, we are still looking for further improvement, so the model C is built.

### Model C: LSTM with as single input



*Figure 7: Model C LSTM with Single inputs*

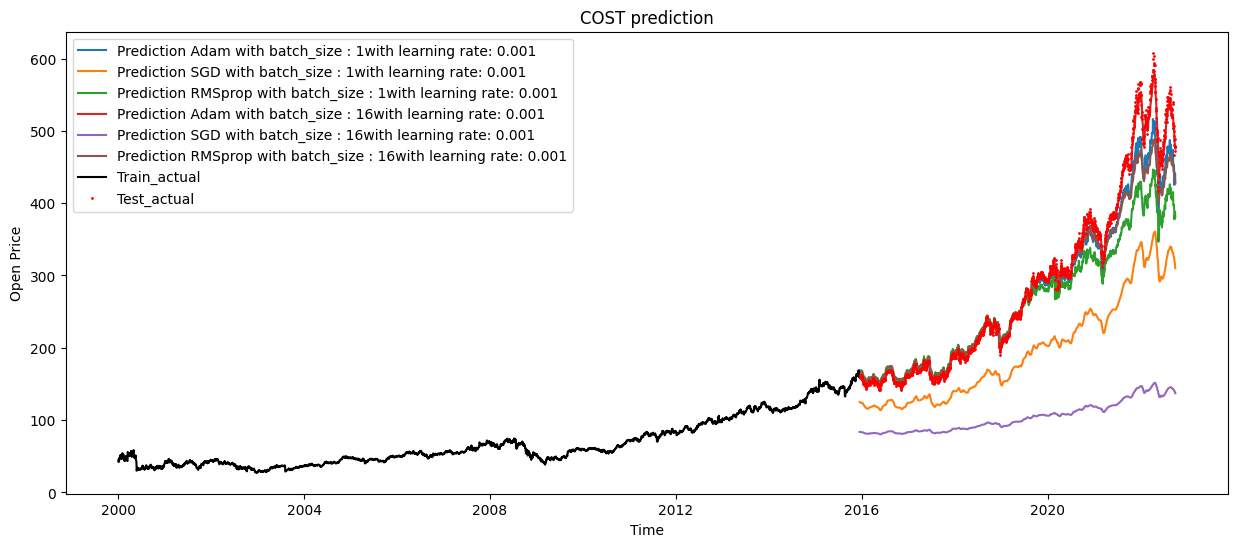
Like Model B, but there is only one branch in Model C, which consists of 2 LSTM layers with return and without return sequences. The output form LSTM layer bypasses to Dense layer with 25 neurons. The LSTM layers and dense layer uses a Rectified Linear Unit (“ReLU”) activation while the Dense output layer is applied linear activation.

This architecture serves as our base model for hyper-parameter tuning, which will be discussed in the next section, since results with relatively higher accuracy were generated. The input of the model is the 3 days close stock price, and the output of the model is the next 1-day stock price.

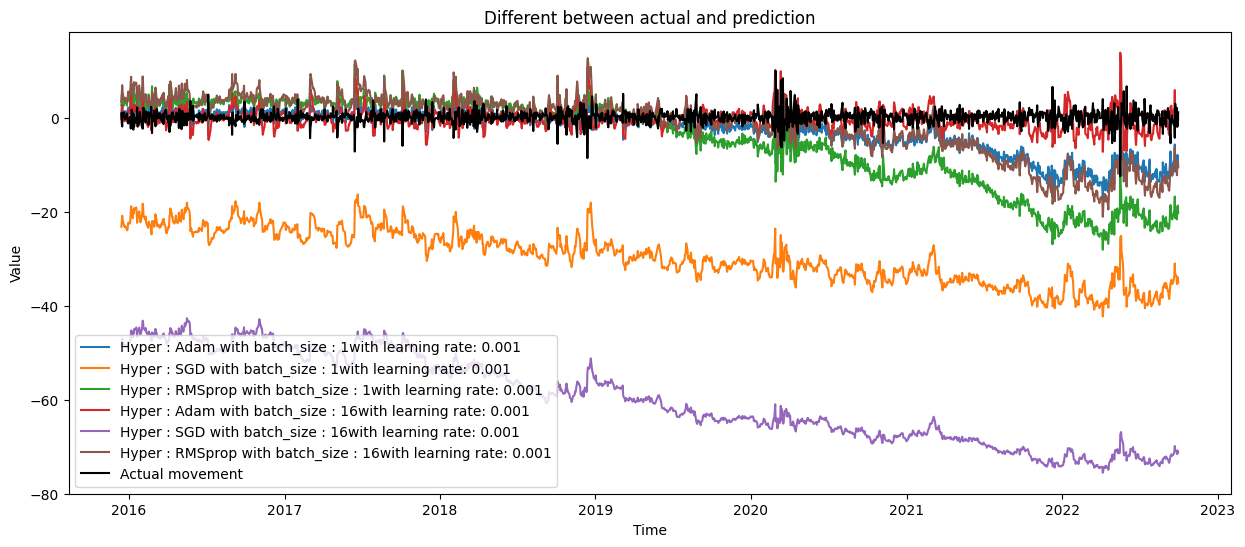
### Model C: Performance evaluation & Hyper-parameter Optimization

|  |  |
| --- | --- |
| https://lh5.googleusercontent.com/teXv4muHwGsN4bXo8C27f9hetqurl6la2RIn0TuboQKVQDYgWfvP1m75dnM6eztn5dWCYG5VDd_jqz5FBl-WXKylsY4ppHGJWOV_K_BHkwju2HDLJnRmpFIkF5n7UmCgZQFd-5HUTfJjrQIGw8o7NtMuvQ=s2048  *Figure 7:* *Training loss against epoch* | https://lh6.googleusercontent.com/rFwYEWbSNQNCRuuGWXIXeaKkrzRcQP-U_76DneXfWAedDFTflaicahyWBl9_pa5RS5ISI4kqKMdfenK7QKsVNBYKhGG0kpBXUZU2057fuEm4YKdLYdseRtt78ceCRz1K6YCM5nijbtbyA7LZlgN2Xmgr_A=s2048  *Figure 7:* *Validation loss against epoch* |

The model is trained by Adam as optimizer. The range batch size tuning is from 1 to 128. The loss function and mean square error for Adam optimizer with 128 batch size can declare normally. In case, the loss function declares in a fast rate, so it is possible to do further tuning in the model parameters or learning rate to reduce the rate of declaring.

*Figure : Costco Prediction*

The lines graph shows the actual price and model predictions with hyper-parameters tuning. The predictions by SGD with 0.001 learning rate do not align with the actual prices. The main reason is the model is vanished in local minimum. The predictions by Adam with learning rate = 0.001 can generally forecast the approximate trend of the stock price.

*Figure : Difference between actual and prediction*

The actual movements of the price and predictions are around 0, but the peaks and turning points of them are not align to each other. Therefore, it reflects that the predictions can catch the general trend and the momentum, instead of the detail feature and daily movement.

# CONCLUSIONS

Deep learning is an advanced method to predict stock price prediction. However, it has its own limitations. In order to extract more information, collecting more data types would be the priority such that a model with high complexity could be built. When we add more input features to the model, it does not improve the prediction accuracy, in contrast to our hypothesis that deep learning model would be able to learn to weigh more towards important features and ignore those features with negligible impact on the prediction accuracy. Therefore, it is concluded that less is more for this project. A simple model with only the historical price of the stock itself performed the best.

Throughout our project, the choice of optimizers plays a significant role in the accuracy of the prediction. Adaptive Moment Estimation Algorithm (“ADAM”) appears to outperform other optimizers, for example Stochastic Gradient Descent (“SGD”) and Root Mean Square Propagation (“RMSProp”) in terms of the speed of convergence and prediction accuracy which proved our hypothesis.

However, even the best model selected with the hyper-parameter tuning, is not able to perform well in the period of high volatility which is characterized by the sharp turnaround. A simple model with historical price is only trying to capture the momentum of price movements of macro-economic environment as well as the cash flow from other markets.

# REFERENCES

[1] Data Collection:

**Stock price:** (consumer staples) (2 days close price) Ticker: WMT // PG // NSRGY // KO // PEP // COST Source: Google Finance

**US Interest Rate (treasury):**  https://home.treasury.gov/resource-center/data-chart-center/interest-rates/TextView?type=daily\_treasury\_yield\_curve&field\_tdr\_date\_value=2022

**US Dollar Index:** https://finance.yahoo.com/quote/DX-Y.NYB

**Credit Spread:** ICE BofA US Corporate Index Option-Adjusted Spread: https://fred.stlouisfed.org/series/BAMLC0A0CM

**CPI:** U.S. Consumer Price Index (CPI) MoM dataset: https://www.kaggle.com/datasets/poormohammadf/us-consumer-price-index-cpi-mom-create-alert

**Unemployment Rate:** https://fred.stlouisfed.org/series/UNRATE