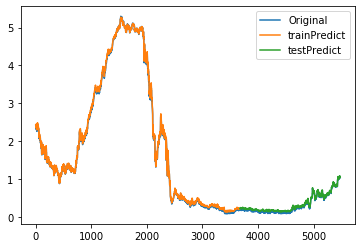
# Findings

## One Time step forecasting with LSTM Recurrent Neural Networks

* Train Score: 0.0474 RMSE
* Test Score: 0.0374 RMSE
* Training set accuracy: 95.29413841664791%
* Testing set accuracy: 84.24805998802185%

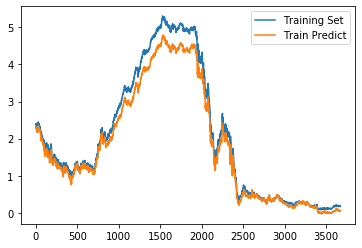


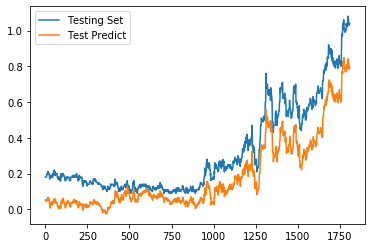
Comments:

* Need to fix randomness for reproducibility, “np.random.seed(7)” somehow doesn’t work
* Need to do more research on the effect of number of epochs (takes about 8s each to run for 15 years of data, now epochs = 10) and batch size (now at 1) when fitting the model
* Any better optimizer? (‘adam’ for now)
* Better testing set accuracy?
* Find a way to interpret RMSE (e.g. range)
* Different parameters (layers of LSTM blocks or neurons) should also be investigated to produce more robust result.

## Multivariate forecasting with LSTM RNN

* Train RMSE: 0.2788
* Test RMSE: 0.1447
* Training set accuracy: 84.10935252904892%
* Testing set accuracy: 48.24609160423279%



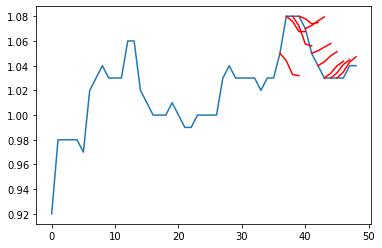


Comments:

* Fix randomness issue with the model
* Testing set accuracy issue
* Research on epochs and batch size. (now at 50 and 72), RMSE decreases to 0.1914 and 0.0268, and accuracy increases to 94.8699% and 93.8627% for training and testing set

## Multi-step forecasting with LSTM RNN

* t+1 RMSE: 0.015218
* t+2 RMSE: 0.023676
* t+3 RMSE: 0.027380



Comments:

* High error, no predictive power. Maybe something wrong with the model set up or code.
* Takes forever to run 15 years of data, now using only 49 most recent data points.
* More experiment can be done on multiple lag timesteps

## Resources:

YouTube Video: Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM)

<https://www.youtube.com/watch?v=WCUNPb-5EYI>

Website: What is the Difference Between a Batch and an Epoch in a Neural Network?

<https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/>

you refer to this guy’s website for knowledge with machine learning. I think it is a great resource!

<https://machinelearningmastery.com/start-here/#timeseries>

# Four Strategies for Multi-Step Time Series Forecasting

* Direct Multi-Step Forecast Strategy
  + Separate model for each forecast time step:
    - prediction(t+1) = model1(obs(t-1), obs(t-2), ..., obs(t-n))
    - prediction(t+2) = model2(obs(t-2), obs(t-3), ..., obs(t-n))
  + Cons:
    - No Opportunity to model correlation between predictions
    - Added computational and maintenance burden
* Recursive Multi-Step Forecast
  + Prediction for the prior time step is used as an input for making a prediction on the following time step
    - prediction(t+1) = model(obs(t-1), obs(t-2), ..., obs(t-n))
    - prediction(t+2) = model(prediction(t+1), obs(t-1), ..., obs(t-n))
  + Cons:
    - Deterioration in accuracy as prediction errors accumulate
* Direct-Recursive Hybrid Strategies
  + Separate models are constructed for each time step with predictions made by models at prior time steps
    - prediction(t+1) = model1(obs(t-1), obs(t-2), ..., obs(t-n))
    - prediction(t+2) = model2(prediction(t+1), obs(t-1), ..., obs(t-n))
  + Pros:
    - Combined benefits of both direct and recursive methods
* Multiple Output Strategy
  + Capable of predicting the entire forecast sequence in one model with ability to learn dependence structure between outputs
    - prediction(t+1), prediction(t+2) = model(obs(t-1), obs(t-2), ..., obs(t-n))
  + Cons:
    - More complex to implement
    - Slower to train and require more data to avoid overfitting