Predicting gold prices with neural networks

Aleksandra Leszczyk

Abstract
We train an LSTM recurrent neural network on historical gold prices to obtain a model to predict future vales.

Keywords
recurrent neural network — LSTM — stock price prediction

Contents

1. Objective

Gold has been an object of people's desire for millennia. Due to its unique color, plasticity, and non-reactivity, it was used for the production of jewelry and currency. Recently, it found application in electronics due to its conductivity properties.

Gold has always been valuable, but its market price constantly changes and fluctuates. Gold price predictions are essential to design optimal production plans for electronic industries. In this work, we construct a long-short-term memory (LSTM) neural network to forecast gold prices on the morning biding session based on the recent price values.

2. Description of data set

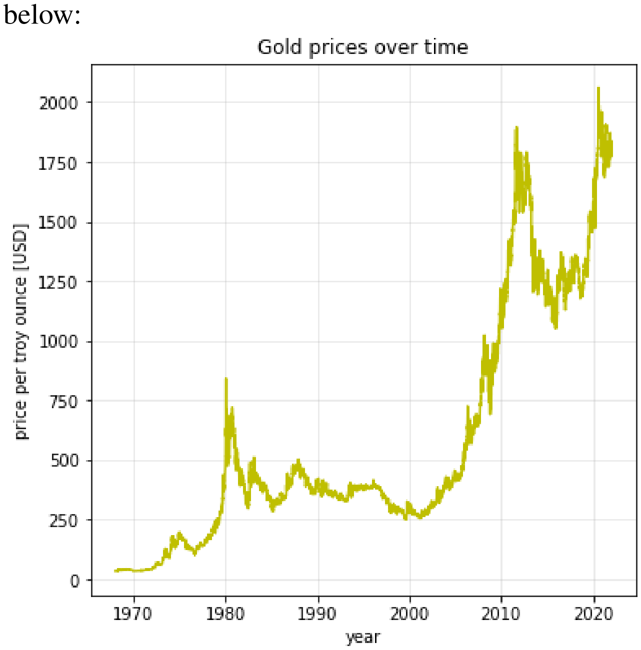
The data set contains a time series of historical gold prices in the London Bullion Market Association (LBMA). The prices are expressed in USD per troy ounce. In this analysis, we focus on morning (AM) biding prices. We retrieved data on 2022/01/23 from Nasdaq using Nasdaq Data Link Python Client technology platform. The data has two columns:

1. Date – date stamp
2. AM - discrete numeric variable expressing gold price

A head of the data is presented in the following table:

Date	AM
1968-01-02	35.18
1968-01-03	35.16
1968-01-04	35.14
1968-01-05	35.14
1968-01-08	35.14

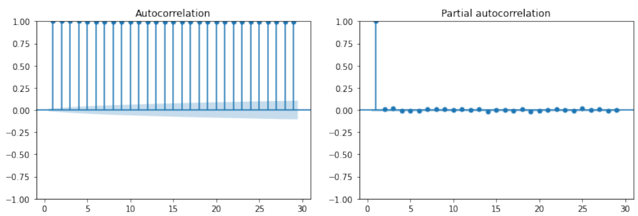
The gold prices are presented on the sequence plot below:



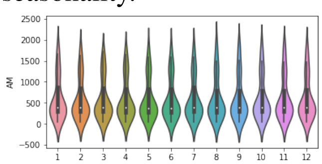
3. Data exploration and cleaning

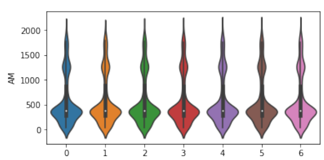
Between 1968-01-02 and 2022-01-21, we have 19743 days, but the data set contains 13663 observations. The data set contains prices from Monday to Friday as LBMA does not work on weekends. The missing values occur on Saturdays and Sundays. We assume that the gold price on Friday is fixed. Thus, we fill missing weekend values with Friday's prices. Also, we used min-max scaling on the gold prices.

We observe non-stationary, self-correlated, and rather non-seasonal structure in the sequence plot. The autocorrelation and partial autocorrelation plots (below this paragraph) confirm these observations.



The following violin plots show data distributions by month and by week. We do not observe significant seasonality.


by month


by week

4. Model and optimization

We use long-short-term memory recurrent neural network with a hyperbolic tangent activation function and the sigmoid activation function to use for the recurrent step. As neural networks are capable to capture sophisticated patterns, we decided to not remove the trend component nor remove the autocorrelations. We tested six models with a different number of cell units (which were equal to 256, 512, 1024, and 2048) and dropout values (0 and 0.2).

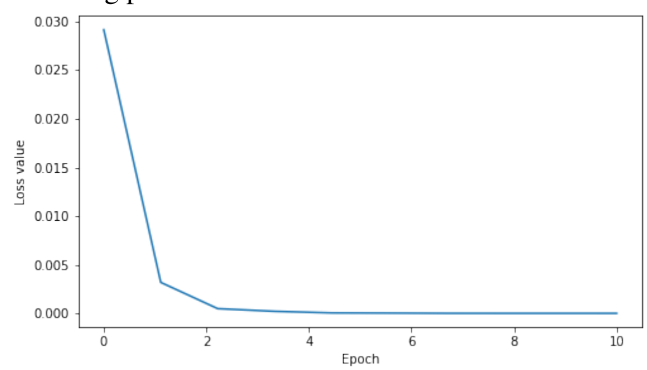
We split data into a training set (1503 samples, each containing gold prices from 31 preceding days) and a test set (154 samples). For weight optimization, we used Adam optimizer with mean squared error as a loss function. We trained our models for 10 epochs.

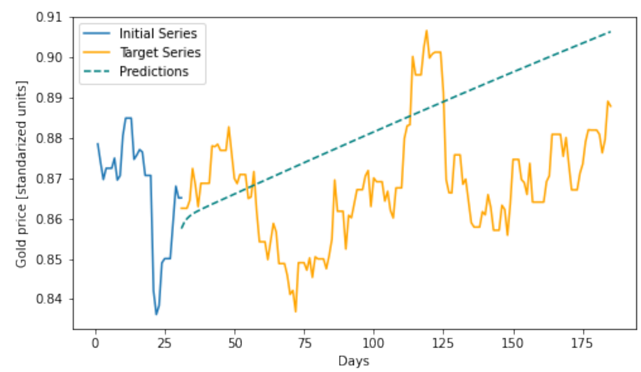
5. Results for different models

The test measures are based on the next-day predictions, so, on the short-term forecast. The mean square errors on train and test data for different models are presented in the following table:

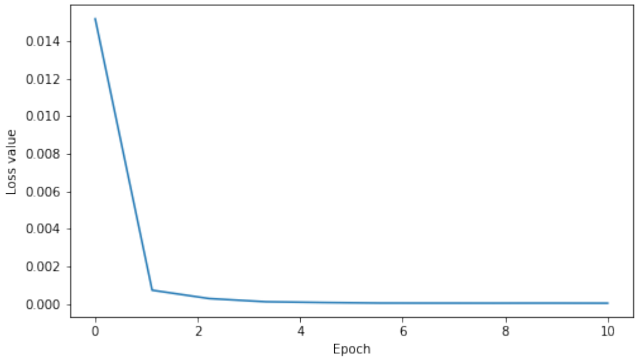
cell units	dropout	MSE (train)	MSE (test)
258	0	4.4E-05	1.5E-02
512	0	3.9E-05	1.5E-02
1024	0	6.5E-05	1.2E-02
2048	0	1.0E-04	1.1E-02
1024	0.2	5.8E-05	1.3E-02
2048	0.2	1.3E-04	9.8E-03

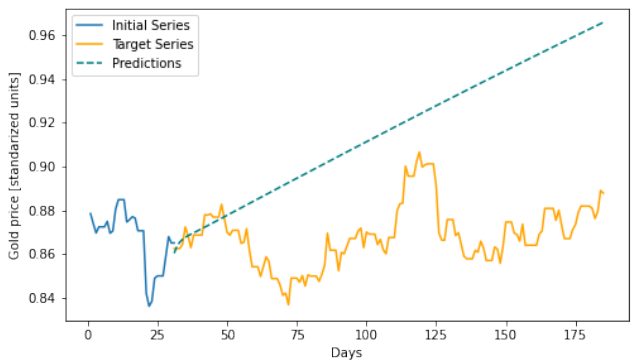
Loss function values along the optimization process and example long-term forecast on test data for the model with 258 cell units and no dropout are presented in the following plots.



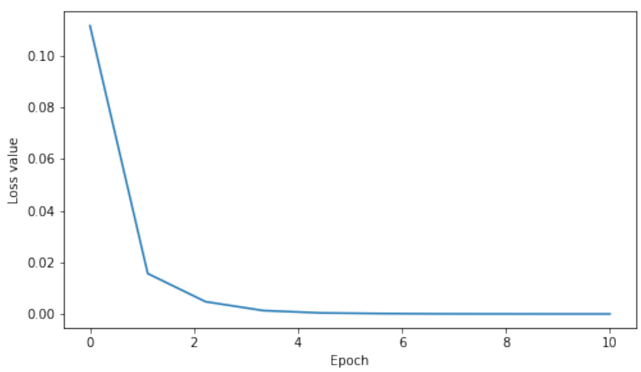


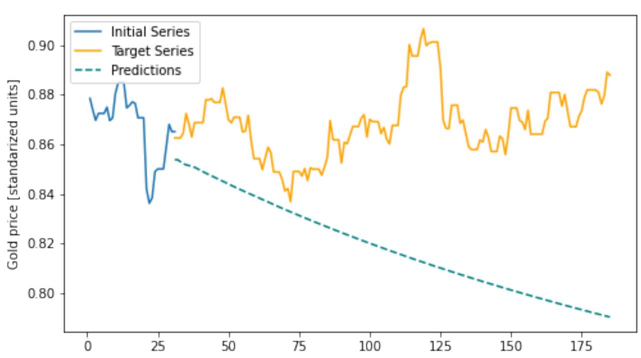
Loss function values along the optimization process and example long-term forecast on test data for the model with 512 cell units and no dropout are presented in the following plots.



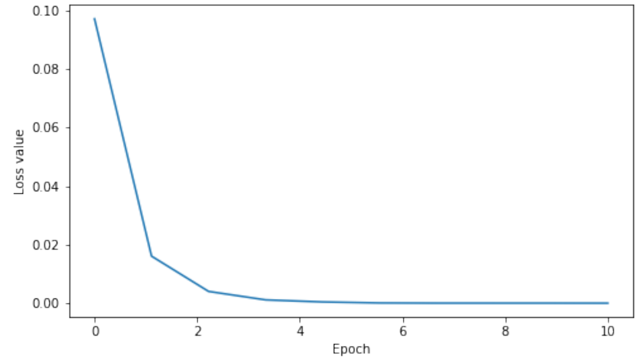


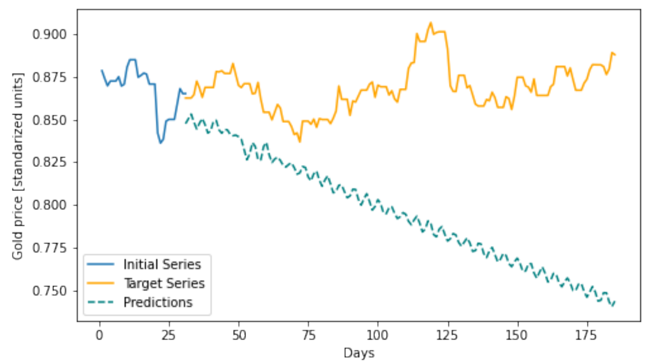
Loss function values along the optimization process and example long-term forecast on test data for the model with 1024 cell units and no dropout are presented in the following plots.



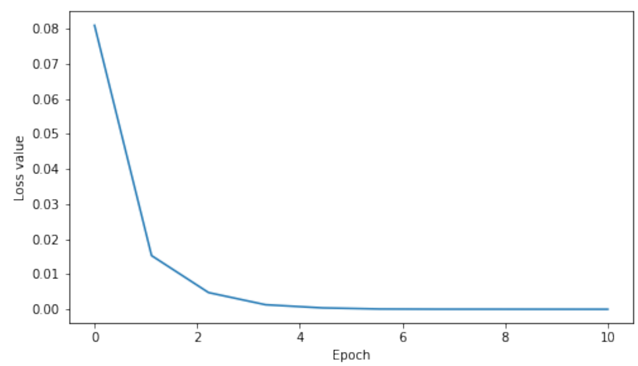


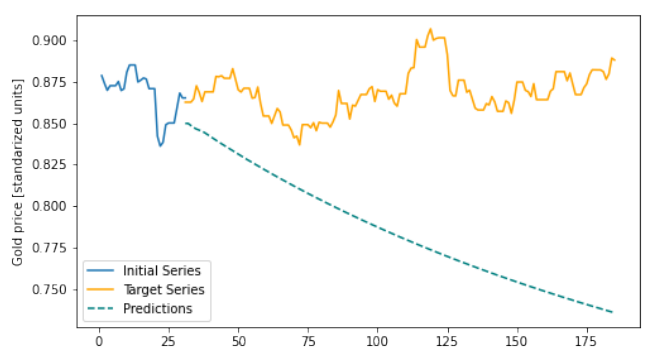
Loss function values along the optimization process and example long-term forecast on test data for the model with 2048 cell units and no dropout are presented in the following plots.



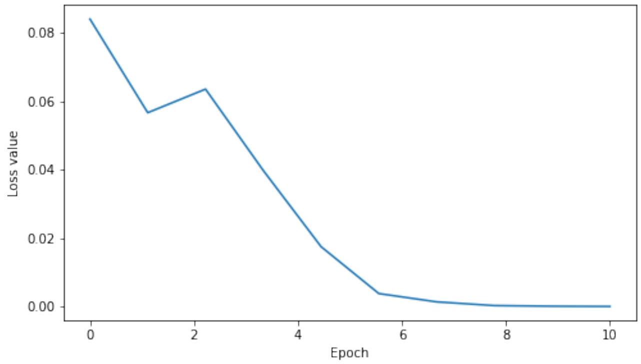


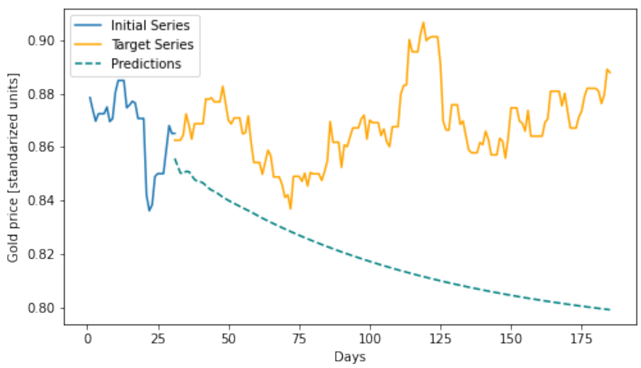
Loss function values along the optimization process and example long-term forecast on test data for the model with 1024 cell units and dropout = 0.2 are presented in the following plots.





Loss function values along the optimization process and example long-term forecast on test data for the model with 2048 cell units and dropout = 0.2 are presented in the following plots.





5.1 Model evaluation

We recommend the LSTM model with 2048 unit cells and 0.2 dropouts. It obtained the highest precision of the next-day forecast as it was indicated by the lowest mean square error on test data. Although long-term forecast for models with simpler structure seems to be closer to the real values, they were generated on one example and cannot be used for accuracy estimation.

6. Summary and key findings

We trained six LSTM neural networks to predict gold prices from the previous observations. As we observe, the higher number of cell units increases the next-day prediction accuracy on test data. Also, regularization (here we used dropout method) helps us to reduce prognosis errors. Most of the models yield smooth long-term forecasts ignoring price fluctuations. Only a model with 2048 cell units and no dropout generated a forecast with a seasonal component. This can be a result of overfitting. The simplest models with 256 and 512 unit cells predicted constant growth of gold prices, while models with a higher number of cell units are more pessimistic and predict a constant decrease.

7. Critical comments and future plans

Although mean square errors indicate that the next-day prediction accuracy is quite good, the long-term forecasts do not capture price fluctuations. This might be due to the fact that fluctuations seem random as compared to previous values so the optimization process removes them. As D. Kahneman (Nobel Prize in Economic Sciences) mentioned in his book *Thinking, Fast and Slow*, the stock exchange prices have features of a completely random process. However, they can depend on how optimistic buyers feel and events in the gold production industry. Thus, the model can be improved by adding these features. The former one can be estimated, for example, with a ratio of optimistic messages with hashtags related to the stock exchange and gold in Twitter. For this purpose, we need to build a bot that collects twits and a neural network trained for recognizing optimistic messages. The second one requires creating a bot that gathers pieces of information on the gold production industry from information services. The information needs to be processed by a similar bot that estimates if the demand and supply of gold in the world rises.