



# Stock Price Prediction with Recurrent Neural Networks

Anran Lu (anranlu)<sup>1</sup> Zeyu Wang (zeyuw)<sup>2</sup> Huanzhong Xu (xuhuanvc)<sup>1</sup>

[1] ICME, Stanford University, [2] Department of Economics, Stanford University

## Abstract

Stock price prediction is an important topic for portfolio construction. Although the Efficient Market Hypothesis developed by Eugene Fama [1] claims that no information can be used to predict the stock market in such a way as to earn greater profits from the stock market, we would like to see if there is a way to beat the market. We use both traditional and neural networks including ARIMA, LSTM, GRU and DA-RNN.

Stock prices are time-series data, hence the natural choice for prediction is RNN. Moreover, stock price today is largely independent of prices one year ago. Therefore, an implicit task for our model is differentiating between “useful” and “unrelated” historical prices, and focusing on remembering historical prices highly correlated with our target. Our algorithm takes historical **adjusted close** as input and outputs predicted price. We compare classical statistic time-series models and deep learning models and evaluate the precision of our models as our metrics.

## Dataset and Feature

The main data we are interested in are S&P500 stocks and SPY itself. All of the historical stock prices are fetched from Yahoo Finance. Since we are trying to make our code as end-to-end as possible, we write our module `data_collecting.py` to fetch the stock prices from Yahoo Finance and save it into a CSV file, helping each user to define his or her own database. User may specify which stock(s) and time period the model should be trained on. The resulting CSV file is saved to the data folder for further training and testing with all five prices including everyday prices such as open, high, and low. Currently the default column we use to develop our model is **adjusted close**, and the default split rate to divide between training set and test(dev) set is **0.9**.

## ARIMA

ARIMA is a traditional statistical model to predict time series. ARIMA is a generalized autoregressive moving average model (ARMA). Especially it is called ARIMA(p, d, q) if  $(1 - B)^d x_t$  is a stationary ARMA(p, q) [2], where B is the backshift operator defined by  $Bx_t = x_{t-1}$ .

## ARIMA (Continued)

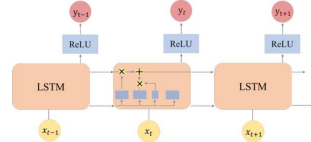
ARIMA(p, q, d) process can be expressed as

$$(1 - \sum_{i=1}^p \phi_i L^i)(1 - L)^d x_t = (1 + \sum_{i=1}^q \theta_i L^i) \epsilon_t,$$

where L is the lag operator,  $\theta$  is the moving average and  $\epsilon$  is the error term.

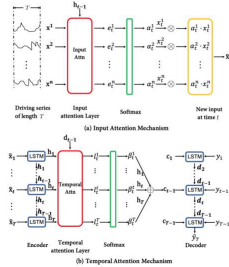
## Recurrent Neural Networks

The two RNN models we try are GRU and LSTM. For the input data, we divide the whole time series into non-overlapping sliding windows, and use a continuous segment of sliding windows to predict the immediately next window after the segment.



## Dual-Stage Attention-Based RNN (DA-RNN)

DA-RNN [3] is adapted from two-stage RNN. An input attention mechanism can select relevant driving series and a temporal attention mechanism can automatically choose relevant encoder hidden states over all time steps. Thus, it can help us to select “useful” input features and capture the long-term temporal dependencies of a time series.



## Results

We evaluate the performance of our methods from the precision of pure stock price and log returns based on MSE loss. Here are our hyper parameter specifications:

Parameter	Value
Window Size	3
Sequence Length	30
Hidden Units	128
Batch Size	8
Num Layers	2

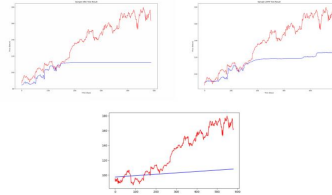


Figure 1: Pure Price Prediction Problems of GRU, LSTM and ARIMA

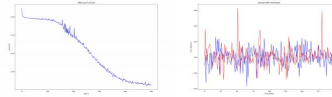


Figure 2: GRU Predicting Log Returns

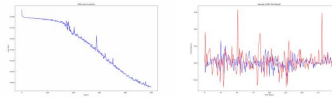


Figure 3: LSTM Predicting Log Returns

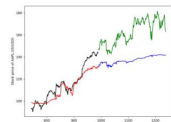


Figure 4: DA-RNN Predicting Stock Price

## Results (Continued)

Table 1: MSE Loss of Different Models

Model	Train Error	Test Error
GRU	0.009311	0.012347
LSTM	0.005251	0.022363

## Discussion

Note that when we try to predict pure stock prices instead of log return, the two RNN models always show a quite high variance, although the loss function indicates a low bias even with various regularization methods such as dropout. An important reason here is that stock prices normally increase monotonically, while the two RNN models cannot predict prices they didn't see in their training sets. Therefore, the two RNN models may better be used to predict log return only.

For DA-RNN, while it has a better performance when predicting stock prices out of the training sets, it still suffers from the same problem when the test set is “too far away” from the training set. In conclusion, RNN-based networks typically have a same deficiency in stock-prediction problems.

## Future Work

- Denoise dataset to make it less ‘uncorrelated’.
- Incorporate the idea of pairs trading where we predict stock price from same ‘sector’ as a group.
- Add other inputs which also impact stock market largely such as economic crisis and earning report, and even unstructured data like news titles.

## Reference

- [1] Eugene Fama. “Efficient Capital Markets: A Review of Theory and Empirical Work”. In: *Journal of Finance* (1970).
- [2] Tze Leung Lai and Haipeng Xing. *Statistical Models and Methods for Financial Markets*. Springer, 2008.
- [3] Yao Qin et al. “A Dual-Stage Attention-Based Recurrent Neural Network for Time Series Prediction”. In: (2017).