TIME SERIES FORECASTING WITH RECURRENT NEURAL NETWORKS

Ondřej Bimka, Josef Kokeš

FS CVUT, ondrej.bimka@fs.cvut.cz

FS CVUT, josef.kokes@fs.cvut.cz

Abstract: The subject of the thesis "Time series forecasting with recurrent neural networks" is the description of the process of training models on typical time series data, the comparison of different models and their use in an application.

Keywords: Machine learning, Neural networks, Data science, Time-series data

1. Introduction

In this work I describe the process of training models on timeseries data predictions using recurrent neural networks (RNNs). I focus mainly on LSTM and GRU RNN cells. I selected the close prices of various cryptocurrencies as the dataset for the thesis. While the crypto market data is quite noisy and difficult to learn, I attempted to create models which would be trained to predict future prices with a degree of accuracy. [1] [2]
The text describes the entire process, from finding suitable data to implementing the best working model into an API. The final API creates predictions on real-time data.

2. Datasets

I used three main datasets. The first one was downloaded, and two others were created with two different python libraries which were wrapped into additional code. The Figure 1 shows the timeline of the three datasets.

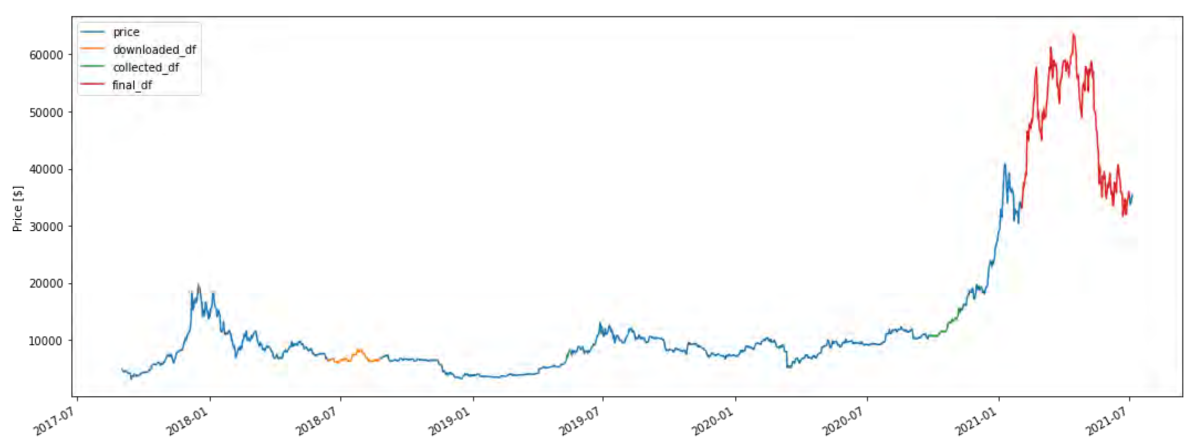

Fig. 1 BTC price with highlighted datasets.

The downloaded and collected datasets were smaller and contained 4 cryptocurrencies (BTC, LTC, ETH, BCH). Final dataset contained 8 highest correlated cryptocurrencies with BTC (BTC, OCEAN, ZRX, ATOM, BNT, ALGO, TWT, SUSHI) from a selected time period.

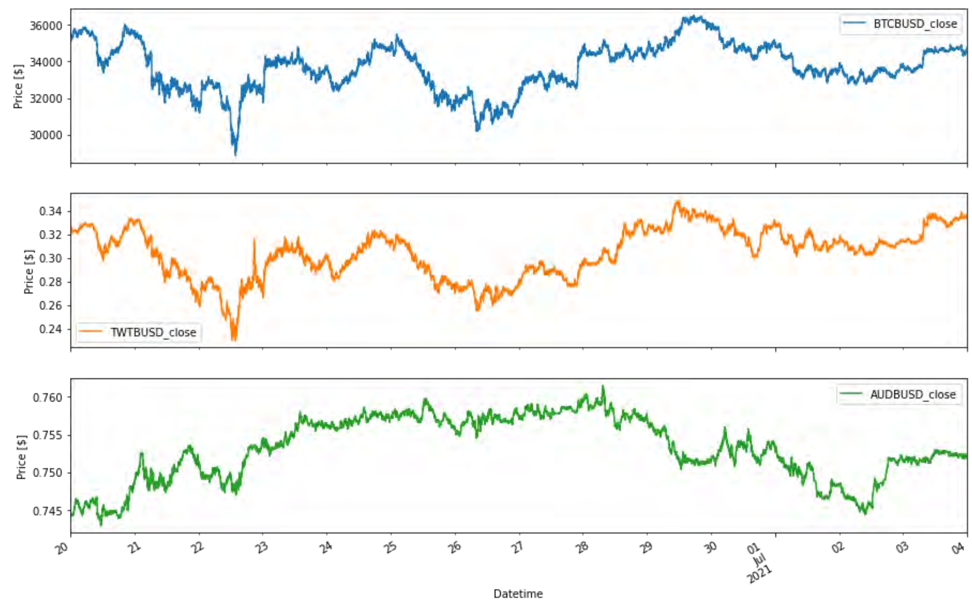

Fig. 2 Example of correlated and uncorrelated data.

3. Data preprocessing

One of the most challenging parts of the whole work was data scaling. The initial approach of data scaling, the z-score only turned out to be quite ineffective. Regardless, I still trained several models using this scaling method and got some interesting results.
Another option was using percentage change + z-score scaling, which worked better overall. The accuracy was higher, and it was easier to implement those models into an API, mostly because scalers worked for longer period on percentage change.

4. Results

The first set of models was trained on the older downloaded dataset using z-score. Predictions were made two minutes into the future using 24 minutes of past data (Fig. 2). The biggest obstacle in the first set was that input data predictions were sometimes shifted. Even in cases where the direction of the prediction was correct the scaling yielded results barely above 50%.

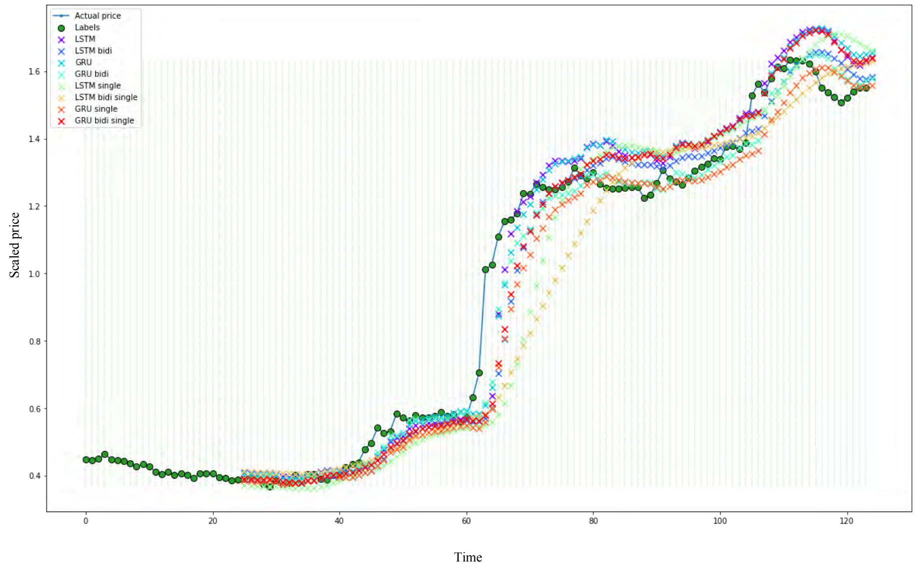

Fig. 3 Example of correlated and uncorrelated data.

In second step I used the downloaded dataset (oldest) with percentage change + z-score to predict 1 minute into the future. This approach worked quite well as on the test part of the downloaded dataset I was able to obtain models with more than 60% accuracy of direction prediction. On the more recent data (beginning of the collected dataset) the accuracy shifted to slightly below 60% accuracy.

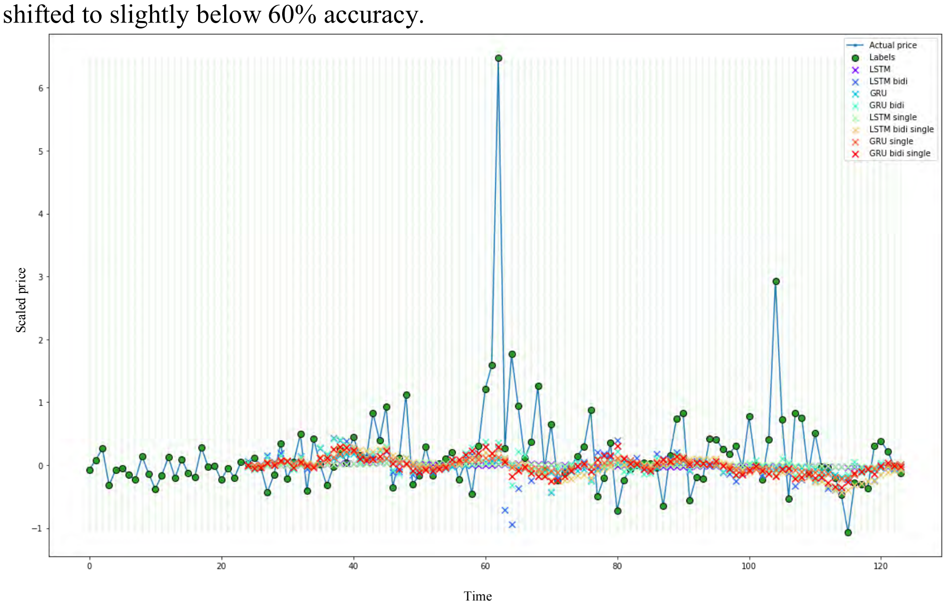

Fig. 4 Example of correlated and uncorrelated data.

The Figure 4 shows clearly that big sudden changes in values are quite unpredictable. That is not a very surprising fact, since models are very careful with more aggressive predictions.
Table 1 lists the results of various architectures with 3 RNN layers, 256 cells and two dense layers first with 32 cells and last with 1 cell. The optimizer in this case was Adam with step 0.01. Adam is one of the most universal optimizers which combines the advantages of AdaGrad and RMSProp. [3]
MAE is Mean Absolute Error score, and Direction tells if at least direction of prediction was correct.

Tab. 1. Performances of various models: Org refers to the test part of the training dataset. New refers to first week of collected dataset. Single means single shot models.

	MAE (org)	Direction (org) [%]	MAE (new)	Direction (new) [%]
LSTM	0.48208	60.19	0.36737	53.65
LSTM - bidirectional	0.45674	56.66	0.3564	52.83
GRU	0.47324	41.94	0.36148	44.35
GRU - bidirectional	0.46807	35.26	0.35103	40.85
LSTM (single)	0.44574	35.26	0.33903	40.7
LSTM - bidirectional (single)	0.44257	59.63	0.33837	53.12
GRU (single)	0.45174	64.74	0.33648	59.3
GRU - bidirectional (single)	0.44074	35.82	0.33681	40.85

The last set of models was trained on the final dataset which contained more data with more cryptocurrencies. When I checked the latest data (end of July 2021) markets became very dynamic, which lead to the fact that I was getting only slightly above 50% accuracy for longer time periods.

5. Predicting API

I created an API which loads selected model and makes predictions every minute based on new incoming data. Data with predictions are stored to MySQL database in predefined interval.

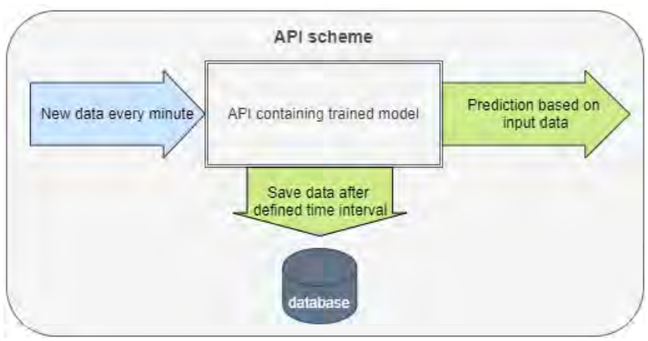

Fig. 5 API scheme.

This allows to load data later and eventually evaluate model performance, which is very useful when models predict in dynamic system such as cryptocurrency market.

6. Conclusions

It is possible to create RNN model which can predict price with limited accuracy. If we consider direction only, I was able to achieve around 60% correct percentage only from close prices on unseen data, which is higher than I expected. Unfortunately, prediction accuracy is closely related to market behavior and model performance can oscillate. For example, for the latest data I was able to get only slightly above 50% accuracy.

Acknowledgement

I wanted to thank everyone who supported me during the preparation and the work on this thesis. I also want to thank my supervisor doc. Ing. Josef Kokeš CSc. for his insides and guidance. Last but not the least, I want to thank my family for providing moral and material support during the course of my studies.

References

[1] S. Hochreiter a J. Schmidhuber, „Long Short-Term Memory," *Neural Computation,* sv. 9, p. 1735–1780, 1997.

[2] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk a Y. Bengio, *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation,* 2014.

[3] D. P. Kingma a J. Ba, *Adam: A Method for Stochastic Optimization,* 2017.

Nové metody a postupy v oblasti přístrojové techniky, automatického řízení a informatiky 2021
New Methods and Practices in the Instrumentation, Automatic Control and Informatics 2021

15. 9. – 17. 9. 2021, Žatec

ISBN 978-80-01-06889-2

Ústav přístrojové a řídicí techniky, FS ČVUT v Praze, Technická 4, Praha 6