RNNs for Bitcoin Price Predictions Case Study

Francesco Peragine
f.peragine@studenti.uniba.it

Department of Informatics
University of Bari Aldo Moro

Abstract *– The continuous escalation of the Bitcoin cryptocurrency market value over the last few years led to an increasing interest to it as investment option on the stock market. This research study aims to illustrate the results obtained by using Machine Learning tools and techniques to predict the Bitcoin market value for several days ahead. In the last section some conclusions and future implementations will be discussed.*

Keyworkds – Bitcoin Price Prediction, Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU), Encoder-Decoder, Bidirectional.

1. INTRODUCTION

The financial word is being increasingly interested in the cryptocurrency market, as it provides a plethora of options for both the public and the private sector. The blockchain ecosystem keeps attracting all kind of users that don't want to miss the new opportunities (FOMO – Fear Of Missing Out) - may them be related to *Non Fungible Tokens* (NFTs), Metaverse or investments - and so on. Even if the cryptocurrency market is far from mature and such price magnitude and volatility were never observed before in the stock market, the public interest keeps rising and new models are constantly being researched to exploit it.

Bitcoin is a decentralized cryptocurrency with no regulation and its daily price changes, moreover, has some features in common with the Stock market. It has its own trends and mechanics and it is inevitably influenced many factors, one of which is politics. By taking into account these considerations, it is possible to build a basic automation tool for prediction. Bitcoin's supply is predetermined *by design* and it is considered deflationary, but the bitcoin itself is almost infinitely divisible and the deflation may not ever occur.

2. LITERATURE

The history of cryptographic currency (*cryptocurrency*) begins in the 1980s started with David Chaum, when he proposed a novel of a cryptographic scheme to blind the content of the message before it is signed so that the signer cannot determine the content. These blind signatures can be publicly verified just like a regular digital signature. Chaum proposed digital cash approach in such a way that is untraceable by another party [1].

The rise of cryptocurrency started on B-money, Wei Dai proposed it [2] as an anonymous and distributed electronic cash system. In that method, he describes two protocols based on network that cannot be traced, where senders and receivers are identified only by their public keys, and each message will be signed by its sender and be opened only by its receiver. Bit Gold In 1998, Nick Szabo [3] propose models a new digital currency, the models based on cryptographic system puzzles, which after being solved, were sent to the Byzantine-fault-tolerant public registry and assigned to the public key of the solver. Hashcash, proposed by Adam Back, is a system relied on a cryptographic hash function to derive a probabilistic proof of computational work as authentication system *Proof Of Work* (PoW) [4].

The last is "RPOW – Reusable Proof of Work", published by Hal Finney in 2004, which proposes currency system based on a *Reusable Proof of Work* protocol [5]

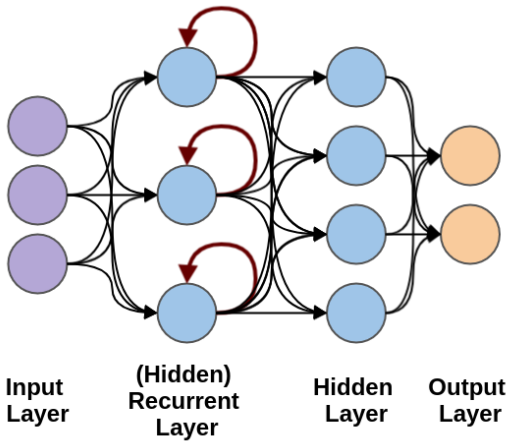
Bitcoin itself was made public as to the first decentralized cryptocurrency with its white paper in 2008 [6].

After the genesis block of the bitcoin protocol was generated, several hundreds of cryptocurrencies were proposed. Besides the Bitcoin protocol literature, the price predictions are also object of researches. Here are reported a few to demonstrate that the research is very active and spread over the use of the recent technologies. Greaves et al. [7] proposed a technique using Logistic Regression and SVM and analyzed using Graph to predict bitcoin price. Huisu Jang et al. [8] published a study based on Bayesian Neural Network. Edwin sin et al. [9] provide topic Bitcoin price prediction using Ensemble of Neural Networks. Arief Radityo et al. [10] proposed a prediction

of bitcoin using Artificial Neural Network technique. They combine with technical market indicators, but the performance results were far from optimal.

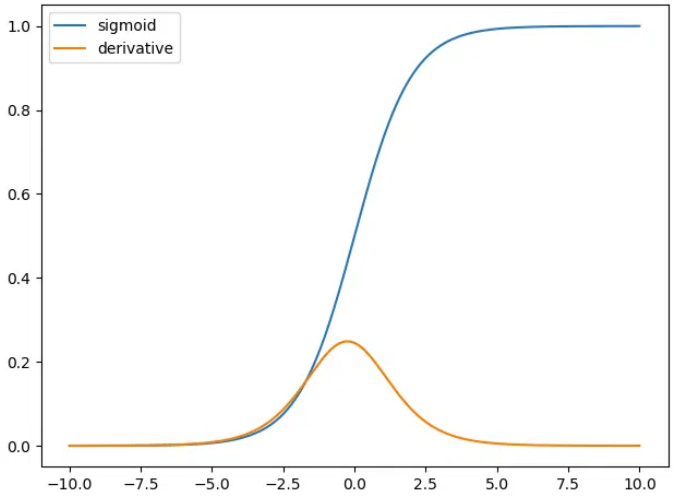
3. MODEL

The idea behind the design of this model is to enable it to process the input without constraints on its length.
To face the problem, the *Recurrent Neural Network* (RNN) model was adopted.
The RNN is a class of artificial neural networks that implements parameter sharing using loops, therefore connections between nodes form a directed or undirected graph along a temporal sequence [11].



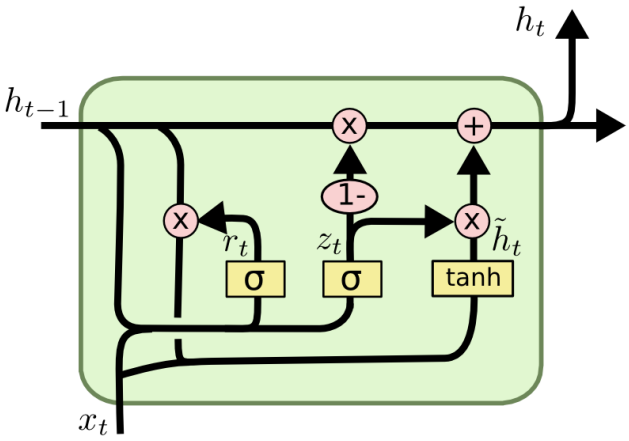
Recurrent Neural Network (RNN)

RNN are designed to work with sequential data, using previous informations to produce the current output. At the last step, the RNN has informations about all the previous steps.
The RNNs, as well as the *Feedforward Neural Networks* (FNNs), suffer from the so called "vanishing gradient problem" [12], that occurs in networks with many layers when the partial derivative of the loss function approaches values close to zero, making the gradient so small as to prevent the weights from changing their values, stopping the network from further training [13].



The sigmoid function and its derivative

The *Long Short-Term Memory* (LSTM) is a variant of the RNN that is able to carry information across many timesteps: like a conveyor belt running parallel to the actual sequence. Information from the sequence can jump onto the conveyor belt at any point, be transported to a later timestep, and jump off, intact, when you need it [14], thus preventing the loss of older signals due to the vanishing gradient problem.
The *Gated Recurrent Unit* (GRU) is the newer generation of the RNN, and the one used in this case study. It is similar to LSTM but has only two additional gates instead of three. GRU merges both the long-term state c_t and the short-term state h_t into one single state vector h_t.
The *update gate* acts similarly to the forget and input gate of an LSTM, as it decides how much information in the hidden state should be updated. The *reset gate* is used to decide how much past information to forget. GRU has fewer tensor operations and therefore is faster to train than LSTM.



Gated Recurrent Unit - GRU

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z)$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r)$$

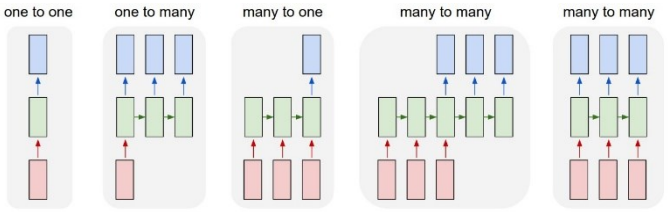
$$\tilde{h}_t = \tanh(W_h \cdot [r_t \odot h_{t-1}, x_t] + b_h)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$$

h_t : hidden layer vectors.
x_t : input vector.
b_z , b_r , b_h : bias vector.
W_z , W_r , W_h : parameter matrices.
σ , tanh : activation functions.

4. METHODOLOGY

The very first rounds of the experimentation have been conducted with a univariate one-to-one model that was able to output a price prediction for a single day. Given the limited utility of that architecture, the model was incrementally developed.



Recurrent Neural Network model architecture

Few hundreds of tests later, the model evolved into a multivariate sequence-to-sequence model based on the Encoder-Decoder architecture.

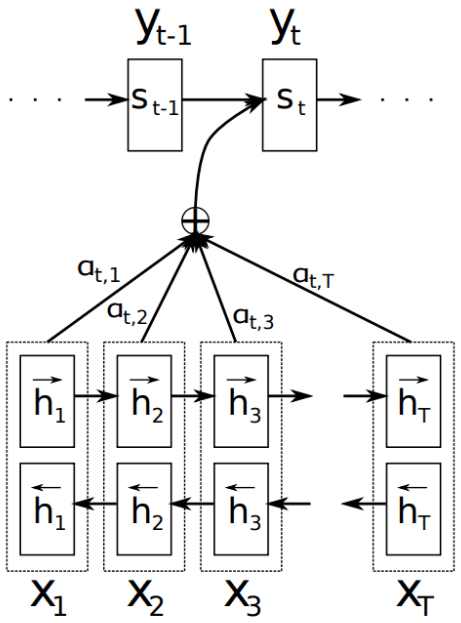
Encoder-Decoder for RNN was presented in the same paper that introduced GRU cells [15].

The model is comprised of two sub-networks: an encoder and a decoder. The *encoder* takes a variable-length sequence as input and transforms it into a state with fixed shape, whilst the *decoder* maps the encoded state of a fixed shape into a variable-length sequence. A *context vector* is responsible for the connection between the two components. The key to the model is that it is entirely trained end-to-end, thus does not train the elements separately.

A potential issue with this architecture is that a NN must compress all necessary informations into a fixed-length vector, resulting in the inability of the system to retain longer sequences of inputs. [16]

The *Attention mechanism* was introduced to cope with this limitation. It's an upgrade of the existing network of *Sequence-to-Sequence models* (Seq2Seq) that is called 'attention' because of its ability to obtain significance in sequences. The core idea is that each time the model predicts an output, it only uses parts of the input where the most relevant information is concentrated, instead of the entire sequence. To rephrase it, it only pays attention to some input sequences. Attention acts as an interface connecting the encoder and the decoder, providing the decoder with information from every encoder hidden state. The attention layer is implemented at the top of the encoder bidirectional RNN layer. The bidirectional layer allows the GRU model to learn the inputs both forward and backwards and concatenates both interpretations [17].

The experimentation showed that, given the relatively shallow model, the attention layer didn't provide any noticeable improvements to the model.



Neural Network Attention mechanism

OPTIMIZER

Empirical tests showed that the RMSprop outperformed both the *Stochastic Gradient Descent* (SGD) and Adam.

LOSS

The performance measures for regression problems are typically selected between Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). RMSE is usually chosen when the distribution resembles a gaussian curve. Given the Bitcoin price nature, the MAE was preferred.

COLLECTION

The process started by collecting the data, which was obtained from Yahoo! Finance historical data prices [17] for the price charts and from Google Trends [18], starting by the 2011.

To study the similarities with the stock market price fluctuations, few technical indicators were introduced (moving averages at 50, 200, 250 days), but the model performances dropped by a big margin, so they were removed.

PREPROCESSING

Some attributes from the gathered data were not considered to be relevant for the training phase and were therefore discarded. After the feature selection phase, only three attributes were kept: *Close value*, *Currency Volume* and *Trend*. This led to the choice of using a multivariate model for both inputs and outputs.

NORMALIZATION

Normalizing a set of data means transforming it in a new set homogenous over all records and fields in a similar scale. Differently from standardization, it is used when the data doesn't have a gaussian distribution and it helps improving data quality by contributing equally to the model fitting and training.
The preferred choice was that of the MinMaxScaler, which scales each feature into a given range of [0,1].

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

ACTIVATION

The choice for the activation function was between the three most popular options: sigmoid, tanh and ReLU.
The *sigmoid* is a squashing function especially useful for models that predict a probability as output, since the probabilities exist in the range of (0, 1). The logistic function can cause a Neural Network to get stuck because of the vanishing gradient problem, moreover it is not centered around zero, so the gradient might be a number too high or too low. The hyperbolic tangent (*tanh*) activation function is still sigmoidal (s-shaped) but its range is between (-1, 1). Its peculiarity is that negative inputs will be mapped as strongly negative, whilst zero inputs will be palled near zero.
The *Rectified Linear Unit* (ReLU) is the most widely used and most recent, and its range is between [0,∞). It has the negative side that negative inputs become zero, decreasing the ability of the model to fit or train properly the data. For the Encoder and Decoder layers the tanh function performed similar results to the ReLU function. For the Dense layer, the ReLU function was chosen because of its positive outputs.

TIME SERIES

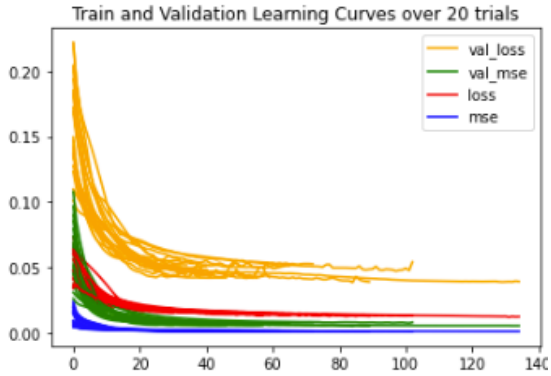
The use of prior timesteps to predict the next ones is called the *sliding window method*, or window method. The sliding window is the basis to turn a time series dataset into a supervised learning problem.
This provides several advantages:

- turn a time series into a regression or classification supervised learning problem for real-valued time series
- apply any linear or non-linear machine learning algorithms if the sequence is preserved
- modify the window size to include more inputs and outputs

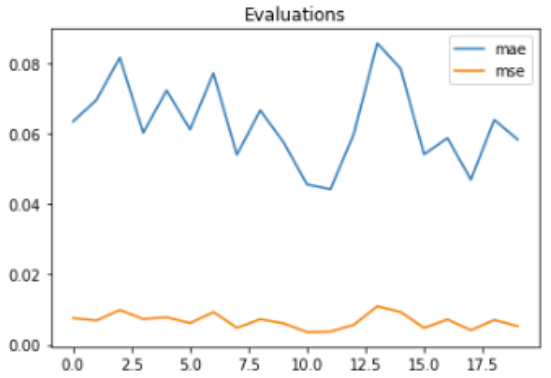
To achieve this, the dataset must be reshaped so that for each sequence of inputs there is a sequence output. Each sequence needs to have the *[samples, timesteps, features]* format.

PERFORMANCE

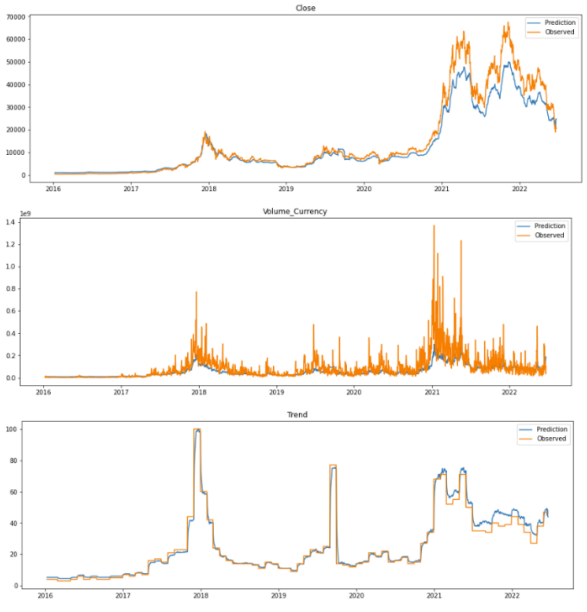
Given the stochastic nature of the algorithms, the evaluations were made by repeating the diagnostic run multiple times, by plotting the train and validation curves to show the behavior of the model and by confronting the results with the test set metrics averages.




```
Metrics overall averages
loss 0.022213695807249655
mse 0.0030264266270680048
val_loss 0.06655659238463273
val_mse 0.014401988219739295
```




```
Tests evaluations overall averages
mae 0.06287569498637818 mse 0.006698416906874627
```



Tests performed with window of 10 inputs and 5 outputs
0.15 Test size, 0.25 Val. Size, lr 0.001, Tanh func, batch size 128

5. CONCLUSION

The Bitcoin price forecasting proved to be a hard task to tackle because different factors:

- The market is new and always growing, there are no past examples in the stock market that follow similar distributions
- enormous oscillations happened in recently in 2022 that greatly altered the validation and the test set evaluations
- multiple factors are to be contemplated and whose data are not easily gathered nor processed: macro/microeconomic factors, political orientations, laws, environment impact, whales' wallets movements, just to cite a few.
- The rest of the cryptocurrencies mostly follow the Bitcoin trend and cannot be used to augment the dataset.

Nonetheless, the proposed model showed interesting results and perspectives.

Future improvements may start by collecting more data and augmenting them, introducing convolutional layers, setting up learning rate decaying methods and furtherly tune the model.

6. REFERENCES

[1] D. Chaum, Blind signatures for untraceable payments., Boston: Springer, 1983.

[2] W. Dai, 1998. [Online]. Available: http://www.weidai.com/bmoney.txt.

[3] N. Szabo, «Secure property titles with owner authority,» 1998. [Online]. Available: https://bitcoinstan.io/prehistory/doc/1998_1.pdf.

[4] A. Back, Hashcash, 1997.

[5] H. Finney, RPOW - Reusable Proofs of Work, 2004.

[6] S. N. a. others, Bitcoin: A peer-to-peer, 2008.

[7] A. G. a. B. Au, Using the bitcoin transaction, 2015.

[8] H. J. a. J. Lee, An Empirical Study on Modeling, 2018.

[9] E. S. a. L. Wang, Bitcoin Price Prediction Using, 2017.

[10] Q. M. a. I. B. A. Radityo, Prediction of Bitcoin exchange rate to American dollar using artificial neural network methods, 2017.

[11] «Recurrent Neural Network Wiki,» [Online]. Available: https://en.wikipedia.org/wiki/Recurrent_neural_network.

[12] T. M. Y. B. Razvan Pascanu, «On the difficulty of training Recurrent Neural Networks,» 2012. [Online]. Available: https://arxiv.org/abs/1211.5063.

[13] S. Basodi, C. Ji, H. Zhang e Y. Pan, «Gradient amplification: An efficient way to train deep neural networks,» [Online]. Available: https://ieeexplore.ieee.org/document/9142152.

[14] F. Chollet, Deep Learning With Python, Manning Shelter Island, 2018.

[15] B. v. M. C. G. D. B. F. B. H. S. Y. B. Kyunghyun Cho, «Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation,» 2014. [Online]. Available: https://arxiv.org/abs/1406.1078.

[16] «Neural Machine Translation by Jointly Learning to Align and Translate,» 2015. [Online]. Available: https://arxiv.org/abs/1409.0473.

[17] M. S. a. K. K. Paliwal, Bidirectional recurrent neural networks, 1997.

[18] «Yahoo! Finance,» [Online]. Available: https://finance.yahoo.com/quote/BTC-USD/history/.

[19] «Google Trends,» [Online]. Available: https://trends.google.it/trends/explore?date=all&q=%2Fm%2F05p0rrx.

[20] «Encoder-Decoder Approaches,» [Online]. Available: https://arxiv.org/abs/1409.1259.