Forecasting financial time series with deep learning

Shvetsov Nikolay [1] **Pilyugina Polina** [1] **Vlasov Andrei** [1] **Talitsky Alexandr** [1]

1. Introduction

The topic of stock prices forecasting has been developing since the first exchange was opened, high volatility of the data and its interdependence on the variety of factors made the forecasting of stock prices an extraordinary tricky task. Several econometric models were developed for this sake, and the increasing amount of papers was written on the application of machine learning algorithms to time series analysis. This project is dedicated to the application of deep learning techniques to the forecasting of stock prices for several selected companies.

2. Literature review

For this project the review of several papers was provided. The focus was set on the most recent papers on the topic. In the work by Adebiyi et al. (2014) the comparison of Neural Networks to econometric models was described, which was the starting point and the biggest inspiration of this project. The author compares deep learning approaches with ARIMA (auto regressive moving average) models on the variety of stock indices in order to validate the application of neural networks to the topic, and the results proved that NNs might be the way to forecast financial time series much more precisely, as the results of NNs were superior to the results of ARIMA.

For the further work lots of papers were analyzed and conferences examined in order to find the most novel and state of the art network architectures for the time series analysis. The focus was then set on the paper dedicated to particular models. Firstly the paper on Bidirectional LSTM was discovered. This network provided promising results and as it had reasonably simple structure it was the first implemented NN for this project. It worked fast and provided quite precise predictions, according to Schuster & Paliwal (1997). This model theoretically involves all training information for making a good prediction on test. It forecasts a trend of data like hidden state model: RNN network learnes particular patterns and recreates them further.

Secondly the Bytenet was implemented, as suggested in the paper by Kalchbrenner et al. (2016). The main feature of this model, that it used 2 neural network: decoder and encoder which causal convolutional layers with dilation.Decoder stacked on the Encoder and generates outputs. Decoder and Encoder work together via 'dynamic unfolding' and in the result it gives us good results for nearest future(10 days), but some ambiguity for wider forecasting period.

Final model that was implemented was Attention is all you need, described by Vaswani et al. (2017). This

3. Data description

Data was retrieved from Yahoo!Finance for the period from 01-Jan-2016 to 28-Mar-2018. This time period was selected in order to compare the results with the existing works, described in papers, and thus evaluate the correctness of our code. In particular, the following indices were selected: DJIA, SP500, NASDAQ, Ebay, Amazon, Apple, Microsoft, Intel. All three models were tested on the data on companies and DJIA and SP500, for DA-RNN as a target NASDAQ was used.

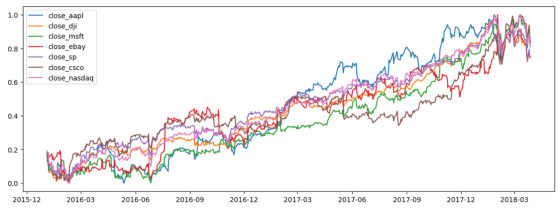


Figure 1. All the time series

In order to include the semantic analysis, the dataset of articles was retrieved from Components. This data contained articles per day for wide period, from which we selected only the sample, corresponding to our chosen period. This dataset contained not only the titles, but also the text of each article, which was then used for the analysis.

4. Methodology

The main ides of this project is to try several neural networks for the forecasting of stock prices for the 40 day period. During the literature review on the topic several best performance models were selected: Bytenet, Attention is all you need and Bidirectional LSTM. This networks were created based on their outlined architecture and were further applied to selected time series of stock prices. The next

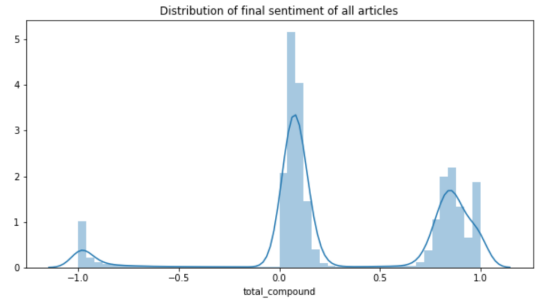


Figure 2. Semantic

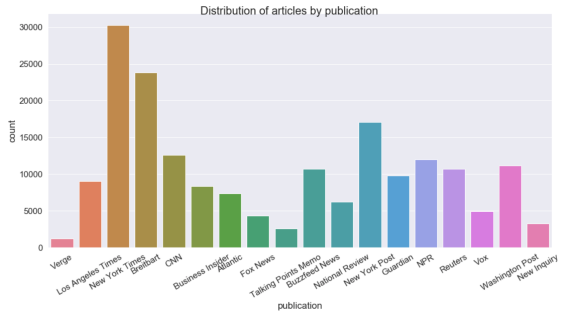


Figure 3. Publicators distribution

step was to chose the best model and evaluate each model performance.

For the further analysis we aimed to add semantic analysis of the news to the Dual-Stage Attention- Based Recurrent Neural Network, as well as the predictions from previous models, in order to predict the NASDAQ time series. Database of news for the selected period was analyzed and added to the feature space, with the lag.

Finally, the comparison of all the models was completed and the baseline regression model was added for the evaluation of the performance. As the result, the project allows to state that deep learning techniques can be successfully applied to the time series forecasting and the precision of this prediction is reasonably high. To be more specific, the best performance was shown by the Attention is all you need network and the addition of semantic analysis to the pipeline increased the quality of prediction.

4.1. Sentiment analysis

Sentiment analysis is an area of study in the field of natural language processing that analyzes opinion of people, their sentiments, attitudes, and emotions via the computational treatment of subjectivity in text.

VADER from nltk was used for this sake, because it has a gold standard list of lexical features (along with their associated sentiment intensity measures) and there is no need to use training data for sentiment analysis. In addition, it works fast and uses emojis, slangs, punctuation and acronyms in sentence for analysis.

For the analysis sentiment of each sentence in each article was calculated and the total compound was calculated, based n the established formula for compound from nltk library. The final compound of an article lied between -1 and 1, indicating positiveness of its content.

In order to use it for further analysis of daily time series, it was not possible to apply the same compound formula for all articles, published during specific date, as it was no longer between -1 and 1 and thus made no sense. The feature creation mechanism was invented for this sake, which calculated histogram-like distribution of articles into 10 bins between -1 and 1, so that the final features were the absolute number of articles in each ”sentiment bin” on each day. Thus we had time series of 10 bins for each day, which we further used in DA RNN.

4.2. Regression analysis

As a baseline model regression model was used. As for the features, lagged polynomial stock prices were added, as well as cosine and sine transformations of lagged prices. Simple linear regression whould have failed to make real prediction, so the transformed prediction was used.

The model was constructed as follows:

1. Stock prices were lagged 5 days before and transformed into features
2. First model was fitted and predictions were made for the further 5 days, for which values of target were known, as lagged target was used in features
3. Target was augmented with this prediction, and new features were calculated based on the predicted ones
4. New model was fitten on the new features
5. The process continues untill we obtain the prediction for 40 days

The results can be found in the figures 4 an 5. It can be seen that the prediction power of the model varies a lot.

4.3. Bidirectional LSTM

First of all, a bi-directional RNN (BRNN) take into account all available input sequence in both directions (the past and future) for estimation of the output vector. For this, input RNN takes the input sequence in a forward time direction and second RNN takes reversed copy of the input sequence in a negative time direction. Inputs of backward states are

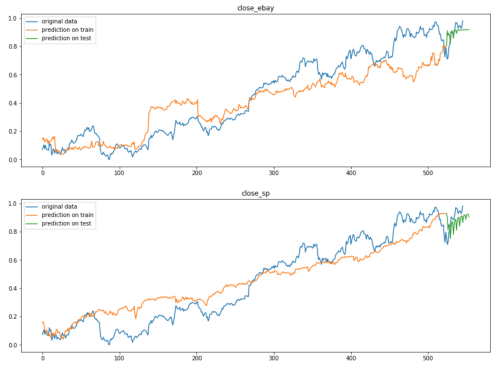


Figure 4. Regression predictions

Figure 5. Regression predictions

not connected with forward states outputs and the two types of neurons have not intersections between them.

LSTM method helps to reducing the effects of exploding or vanishing gradients during the training step. Finally, the model learn long-short term sequential dependencies in data.

For BiLSTM, we considered last 5 stock's close value and predict next one. We don't use additional information from test data for model training.

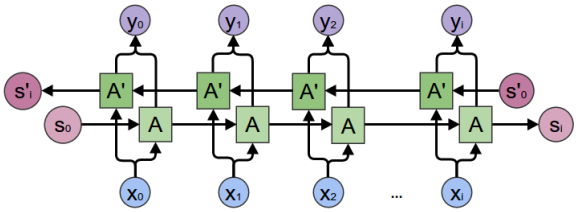
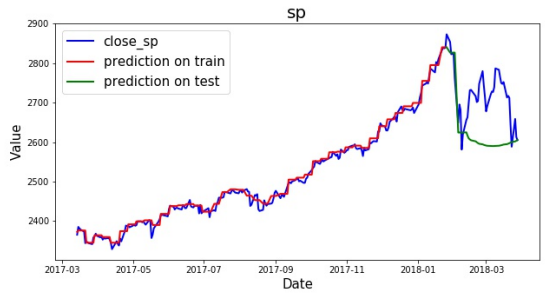
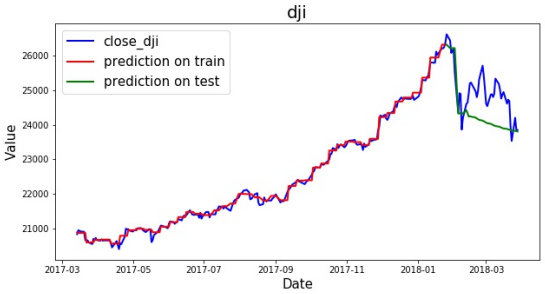


Figure 6. BiRNN Architecture

From results, we can conclude that BiLSTM make a good prediction when data has a trend like for indexes. However,if we make stock prediction, our model has bad results.

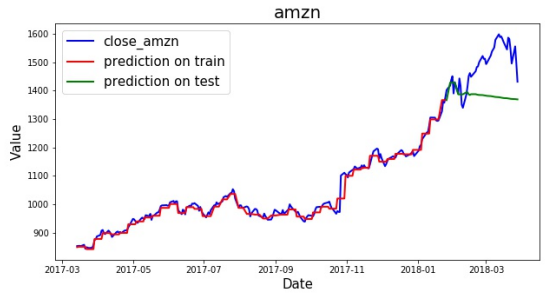


a)

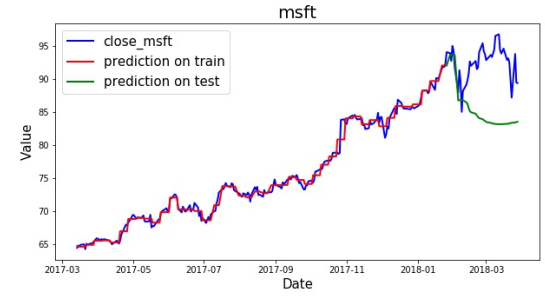


b)

Figure 7. Results of BiLSTM for indexes SP a) 500 b) DJ index



c)



d)

Figure 8. Results of BiLSTM for stocks c) Amazon d) Microsoft

4.4. Bytenet

ByteNet is neural network for machine translation which will be present in 2017 in Neural Machine Translation in

Linear Time by Nal Kalchbrenner, Lasse Espeholt[1] The ByteNet is a one-dimensional convolutional neural network that composed of two parts: Encoder and Decoder.

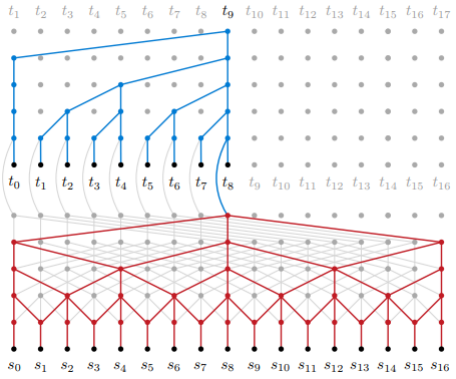


Figure 9. ByteNet Architecture

The Encoder is red neural network on picture 1, is formed from one-dimensional causal convolutional layers that use dilation.Convolution use dilation to increase receptive field of the Decoder. The dilation makes the receptive field grow exponentially in terms of depth of the network. The decoder also consist from one-dimensional convolutional layers with dilation and they are masked. The decoder stacked on the source network and generates variable-length outputs.Encoder and decoder network that process sequences of different lengths cannot be directly connected due to the different sizes of the computed representations. ByteNet authors solve this problem via 'dynamic unfolding'. The decoder applies masked 1-d convolutions to the input embedding tensor that have a masked kernel of size k. The masking ensures that information from future tokens does not affect the prediction of the current token. Each layer is wrapped in a residual block that contains additional convolutional layers with kernel size 1. In this work was applied residual blocks with ReLU activation function.

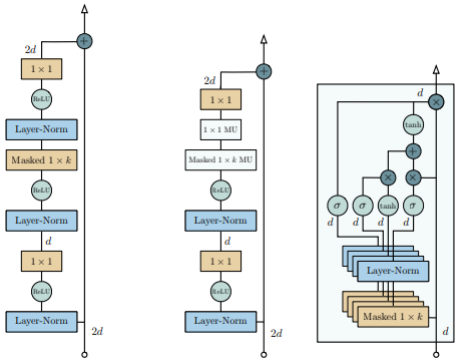
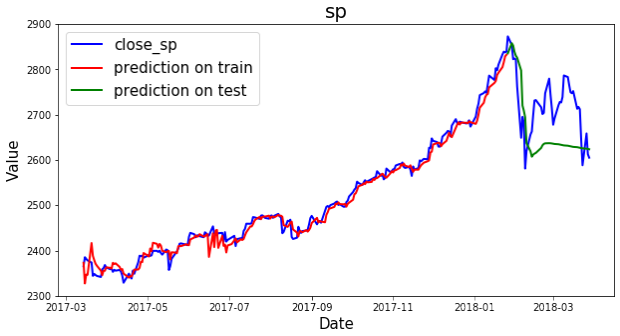
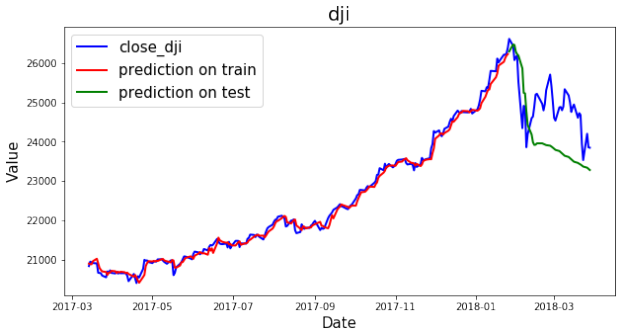


Figure 10. Residual Blocks

In this work we apply ByteNet in which way, we took last 5 stock's close value and predict next one, as a results we can achieve good prediction on train data and on first 10 days, for next 30 days neuronet gave bad prediction and start predict some constants. for some data it works better.



a)



b)

Figure 11. Results of ByteNet a) SP 500 b) DJ index

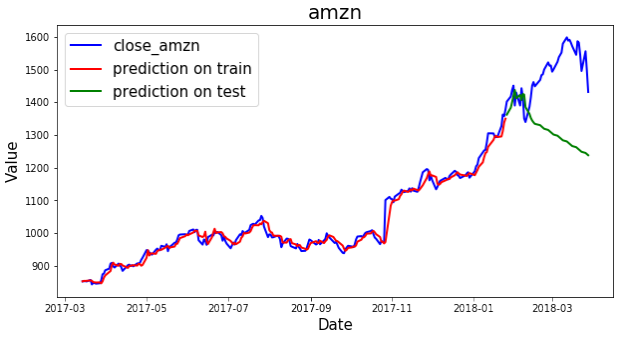
4.5. Attention is All you need

Most competitive neural sequence transduction models have an encoder-decoder structure. This network also has the same structure.

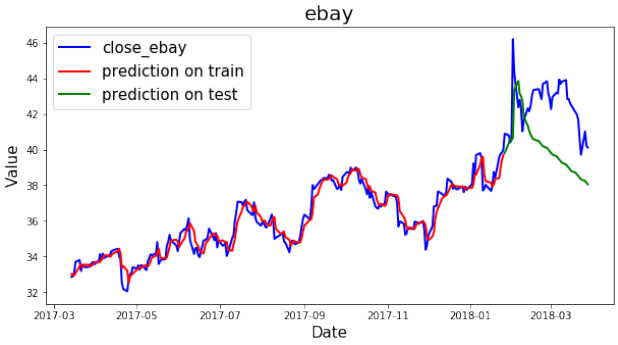
Encoder: The encoder is composed of a stack of N = 6 identical layers. Each layer has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a simple, position wise fully connected feed-forward network.

Decoder: The decoder is also composed of a stack of N = 6 identical layers. In addition to the two sub-layers in each encoder layer, the decoder inserts a third sub-layer, which performs multi-head attention over the output of the encoder stack.

Scaler dot product attention: We call our particular attention "Scaled Dot-Product Attention" (Figure 2). The input consists of queries and keys of dimension d_k, and values of

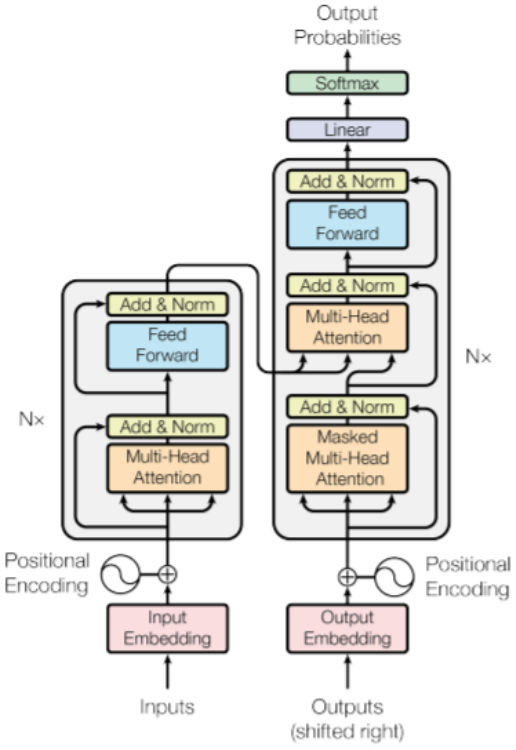


c)

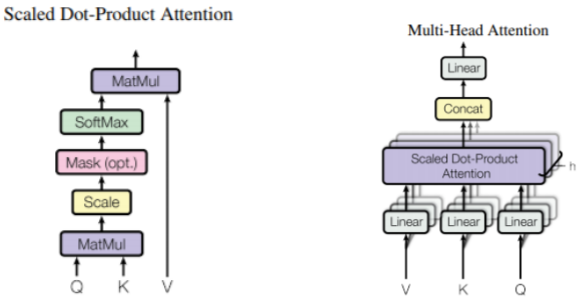


d)

Figure 12. Results of ByteNet a) Amazon b) Ebay



dimension d_v. We compute the dot products of the query with all keys, divide each by $\sqrt{d_k}$



$$Attention(Q, K, V) = softmax(\frac{QK^\top}{\sqrt{d_k}})V$$

Multihead attention Instead of performing a single attention function with d model-dimensional keys, values and queries, we found it beneficial to linearly project the queries, keys and values h times with different, learned linear projections to d_k, d_k and d_v dimensions, respectively. On each of these projected versions of queries, keys and values we then perform the attention function in parallel, yielding d_v-dimensional output values. Multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions. With a single attention head, averaging inhibits this.

$$Multihead(Q, K, V) = Concat(head_1, ..., head_h)W^0$$

Where $head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$ and W_i is some learnable parameters.

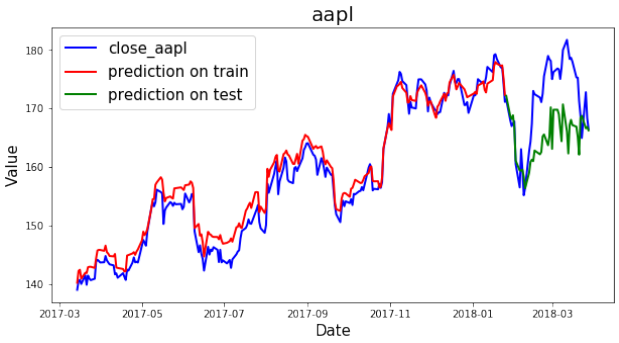
Results

Results presented in 13 and 14. This network is the best for direct time series forecasting. It shows great results, but it also has the largest training time between our networks. For some prediction we got almost the same results, but for another one we got wrong prediction. But the best companies, such as Microsoft, Apple and others, are presented in plots. and we can view excellent matching.

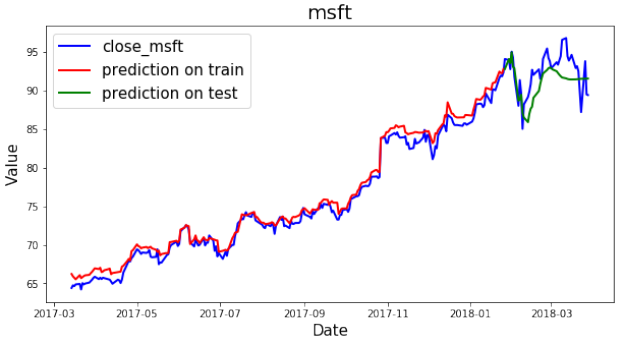
4.6. Dual-Stage Attention-Based Recurrent Neural Network

Dual-stage attention-based recurrent neural network (DA-RNN) is used for time series forecasting. It has input as n driving time series and gets prediction for target. In our case the inputs are predicted time series from Attention-is-all-you-need network for all companies, what we have. And target DA-RNN is NASDAQ index.

Encoder Firstly we apply input attention mechanism followed LSTM as encoder.

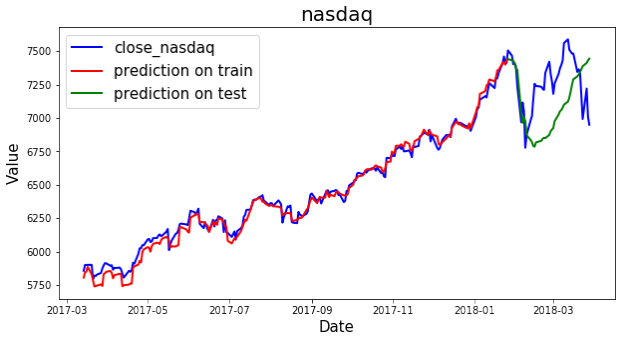


a)

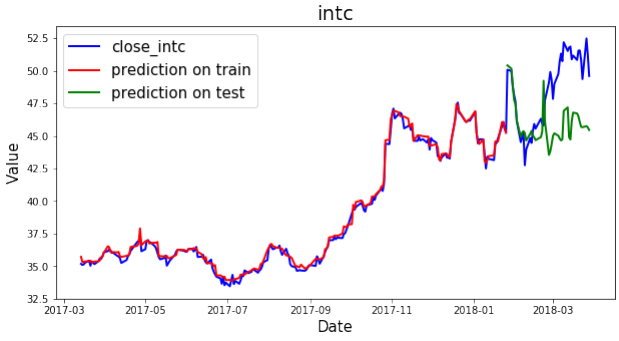


b)

Figure 13. Results of Attention-all-you-need a) Apple b) Microsoft



a)



b)

Figure 14. Results of Attention-all-you-need a) Nasdaq index b) Intel

Decoder To predict target, we use another LSTM layer with temporal attention layer. In more details you can read in the article (Qin et al.)

Results Using news and predicted Stock Prices we get a excellent results and for test part we got almost the same results. In more details you can view in attached code.

References

Adebiyi, A. A., Adewumi, A. O., and Ayo, C. K. Comparison of ARIMA and Artificial Neural Networks Models for Stock Price Prediction. *Journal of Applied Mathematics*, 2014:1–7, mar 2014. ISSN 1110-757X. doi: 10.1155/2014/614342. URL http://www.hindawi.com/journals/jam/2014/614342/.

Che, Z., Purushotham, S., Cho, K., Sontag, D., and Liu, Y. Recurrent Neural Networks for Multivariate Time Series with Missing Values. *Scientific Reports*, 8(1): 6085, dec 2018. ISSN 2045-2322. doi: 10.1038/s41598-018-24271-9. URL http://www.nature.com/articles/s41598-018-24271-9.

Fu, R., Chen, J., Zeng, S., Zhuang, Y., and Sudjianto, A. Time Series Simulation by Conditional Generative Adversarial Net. Technical report. URL www.macroadvisers.com.

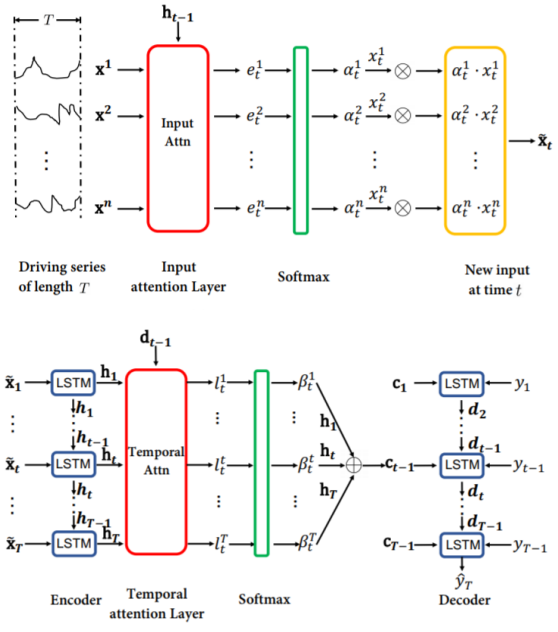


Figure 15. Dual Stage Network: a) Encoder b) Decoder

Kalchbrenner, N., Espeholt, L., Simonyan, K., van den Oord, A., Graves, A., and Kavukcuoglu, K. Neural Machine

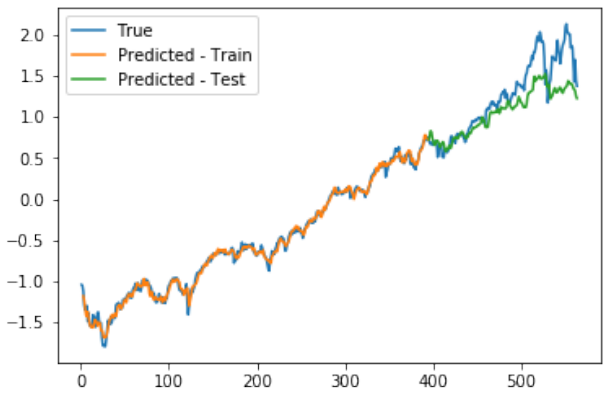


Figure 16. Prediction NASDAQ index of Dual Stage Attention Network

Translation in Linear Time. oct 2016. URL http://arxiv.org/abs/1610.10099.

Karim, F., Majumdar, S., Darabi, H., and Chen, S. LSTM Fully Convolutional Networks for Time Series Classification. *IEEE Access*, 6:1662–1669, 2018. ISSN 2169-3536. doi: 10.1109/ACCESS.2017.2779939. URL http://ieeexplore.ieee.org/document/8141873/.

Qin, Y., Song, D., Chen, H., Cheng, W., Jiang, G., and Cottrell, G. W. A Dual-Stage Attention-Based Recurrent Neural Network for Time Series Prediction. Technical report. URL https://arxiv.org/pdf/1704.02971.pdf.

Schuster, M. and Paliwal, K. K. Bidirectional Recurrent Neural Networks. Technical Report 11, 1997. URL https://pdfs.semanticscholar.org/4b80/89bc9b49f84de43acc2eb8900035f7d492b2.pdf.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention Is All You Need. jun 2017. URL http://arxiv.org/abs/1706.03762.

Zhang, L., Aggarwal, C., and Qi, G.-J. Stock Price Prediction via Discovering Multi-Frequency Trading Patterns. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '17*, pp. 2141–2149, New York, New York, USA, 2017. ACM Press. ISBN 9781450348874. doi: 10.1145/3097983.3098117. URL http://dl.acm.org/citation.cfm?doid=3097983.3098117.

A. Appendix A

Nikolay did the whole implementation of Bytenet. He recreated the model from the paper Kalchbrenner et al. (2016), including residual blocks construction. He ran a lot of tests on different financial datasets and corrected the sizes of the model layers in order to be used for our purposes. In addition, he worked on the correction of the DA RNN model: he adjusted input sizes and tested whether the inclusion of news helped the model improve.

Alexander implemented the Attention Is All You Need network from scratch as it was described in the paper (Vaswani et al., 2017). He has also corrected layer sizes to much the purposes of our study. He worked on the code refactoring of all models and implemented the summary file of all the models. Additionally he worked on the correction of DA RNN as well. Also he ran experiments with data, tried different time series and sizes, in order to find the best one.

Andrey worked on the implementation of the Bidirectional LSTM from scratch based on the paper Schuster & Paliwal (1997). He ran the tests of this model on the data. Also he worked on the implementation of the semantic analysis and created the script for parallel computation of the sentiment for each article. He additionally run experiments to test the use of semantic analysis on the data.

Polina worked on the data preprocessing and gathering, outlining the main plan and possible solutions. She worked on some bug fixes in Bytenet and DA RNN models. She also worked on the implementation of DA RNN and its correction, basically rewriting the training and preprocessing code for our case. She worked with Andrew's semantic results in order to use them in the final model and created bin-like features for the analysis. She also created and tested baseline methods like ARIMA and regression models.

B. Appendix B

https://github.com/Zhenye-Na/DA-RNN https://github.com/Seanny123/da-rnn This two repositories were used in the creation and the correction of DA-RNN for the NASDAQ prediction.

https://towardsdatascience.com/basic-binary-sentiment-analysis-using-nltk-c94ba17ae386 From here we have taken ideas for the semantic analysis

https://towardsdatascience.com/intuitive-understanding-of-attention-mechanism-in-deep-learning-6c9482aecf4f Description of Attention mechanism