Stevens Institute of Technology

FE800

Special Problems in Financial Engineering

Stock price forecasts with Recurrent Neural Networks

Authors:
Sarp Uslu, Jingyi Ma

Advisor:
Dr. Khaldoun Khashanah

May 10, 2018

Abstract: In this project we investigate the prediction error of Recurrent Neural Networks (RNN) for single day ahead stock prices. We examine different RNN architectures and various input variables besides the price data such as clustered indicator signals. Furthermore, we investigate signal processing techniques such as the Empirical Mode Decomposition for pre-processing the price data.

1 Introduction

Time series forecasting is a widely discussed topic that has applications in many fields especially in finance. Researchers have applied various methods based on different disciplines such as statistics, physics, machine learning and signal processing. In this project, our main focus is daily stock price forecasting with recurrent neural networks. Additionally, we have experimented with methods such as Empirical Mode Decomposition and clustered indicator signals to see whether forecasts can be improved.

Independent of the method used in forecasting, methodology remains the same. The goal is to approximate a function between past and future data points. In this project we solely focus on one step ahead predictions therefore the function we would like to approximate is in the following form for the univariate case:

$$x_{t+1} = f(x_t, x_{t-1}, ..., x_{t-p}) \tag{1}$$

For the multivariate case:

$$x_{t+1} = f(x_t, x_{t-1}, ..., x_{t-p}, y_{1,t}, y_{1,t-1}, ..., y_{1,t-p}, y_{a,t}, y_{a,t-1}, ..., y_{a,t-p}) \tag{2}$$

where x_t is the daily price series and y_t is external time series. a is the number of exogenous variables and p is the number of lags.

Multi-Layer Perceptrons (MLP) are universal function approximators and they could be used for approximating the function described above using a fixed-length window time series embedding.[1] While this approach showed promising results, a better approach is to make the network recurrent by using internal feedback loops. This way the neural network has an internal representation of memory and it preserves the temporal context of the input sequence. Recurrent Neural Networks (RNN) has been used in many sequence based recognition and generation tasks.[2]

1. Salvatore Marra and Francesco C Morabito, "A New Technique for Solar Activity Forecasting using Recurrent Elman Networks," *Computational Intelligence* 3, no. 1 (2006): 8–13.
2. Martin Längkvist, Lars Karlsson, and Amy Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," *Pattern Recognition Letters*, 2014, ISSN: 01678655, doi:10.1016/j.patrec.2014.01.008.

2 Recurrent Neural Networks

RNNs are feed-forward neural networks where outputs of neurons are connected to their inputs. Outputs at each time step depend on previous inputs and computations. This architecture creates an internal memory structure and eliminates the need of fixed-length window time series embedding.[3] In other words, single lag time delay is sufficient to model a dynamic process. However, some researchers still suggest the use of these time-delayed connections especially when long-term dependencies matter.[4]

There are various ways to increase the memory capacity of RNNs. A basic approach is to increase the depth of the network by using multiple layers. Another solution is to build the network using special neuron cells which are designed to internalize long-term dependencies such as Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) cells.[5]

While there are ways to increase the internal memory of the RNN, it is not clear whether a larger or a deeper network will lead to better performance, at least for the task of forecasting stock prices. The best approach is to start with a small network and increase the complexity until no more improvements are observed. Furthermore, it is difficult to know in advance which network will perform well for a given task.[6]

Elman Neural Network, also called simple RNN, is the most basic implementation of the internal memory feedback structure. LSTM and GRU networks are just variations of the simple RNN. For our task, simple RNN training was unstable and larger networks were required to reach same level of prediction accuracy compared to LSTM and GRU.

LSTM and GRU cells are specifically designed for learning long term dependencies. However, this superiority doesn't necessarily translate to simpler tasks.[7]

3. Mohammad Assaad, Romuald Boné, and Hubert Cardot, "LNCS 4233 - Predicting Chaotic Time Series by Boosted Recurrent Neural Networks," *Part II LNCS* 4233 (2006): 831–840.

4. Romuald Bonã, Michel Crucianu, and Jean-Pierre Asselin De Beauville, "Learning long-term dependencies by the selective addition of time-delayed connections to recurrent neural networks," *Neurocomputing* 48 (2002): 251–266, `www.elsevier.com/locate/neucom`.

5. Filippo Maria Bianchi et al., "An overview and comparative analysis of Recurrent Neural Networks for Short Term Load Forecasting":1–41.

6. Muhammad Ardalani-Farsa and Saeed Zolfaghari, "Chaotic time series prediction with residual analysis method using hybrid Elman-NARX neural networks," *Neurocomputing*, 2010, ISSN: 09252312, doi:`10.1016/j.neucom.2010.06.004`.

7. Ibid.

3 Clustering by Self Organizing Maps

Self Organizing Maps (SOM) is an unsupervised learning and clustering algorithm. Original SOM method is typically used for making a nonlinear projection of high dimensional data into a two dimensional grid and it provides great visualizations of the mapping that takes place.

Clusters are represented by a neuron, a vector of weights w_i, and each input vector $x(t)$,where t is time index, is mapped to a cluster by a similarity measure. In other words an input vector is associated with the best matching unit, called a winning neuron, which has the smallest distance $min_i||x(t) - w_i||$, where i is the index of cluster. Winning neuron and its neighbors' weights are updated proportionally as each input vector is presented to the network.[8] The learning rule can be represented as follows:

$$\Delta w_i = \gamma h_{ik}(x(t) - w_i)$$

where γ is the learning rate and k is the index of the winning neuron, and h_{ik} is the topological neighborhood function.

$$h_{ik} = e^{\frac{-||r_i - r_k||^2}{\sigma(t)^2}}$$

where r_i is the map coordinate of neuron i, $\sigma(t)$ is a decreasing standard deviation with respect to time.[9]

Just like the RNN is a temporal extension upon MLP. Recurrent SOM is an extension upon the original SOM and it is intended to be used for temporal sequences.[10]

SOM is robust to parameter selection and produce better visualizations compared to hierarchical an k-means clustering.[11]

8. Aymen Cherif, Hubert Cardot, and Romuald Boné, "SOM time series clustering and prediction with recurrent neural networks," *Neurocomputing*, 2011, ISSN: 09252312, doi:10.1016/j.neucom.2010.11.026.
9. Barbara Hammer et al., "Self-Organizing Maps for Time Series," *WSOM 5th Workshop On Self-Organizing Maps*, 2005, 1–8, doi:1011619893, http://www.math.unipd.it/~sperduti/PAPERI/wsom2005.pdf%5Cnpapers3://publication/uuid/EB621581-B75E-494C-96B3-7C0C0418DFA4.
10. Thomas Voegtlin, "Recursive self-organizing maps," *Neural Networks* 15, nos. 8-9 (2002): 979–991.
11. X Wang et al., "A scalable method for time series clustering," *Unrefereed research papers* 1 (2004), papers://b6c7d293-c492-48a4-91d5-8fae456be1fa/Paper/p3095%5Cnfile:///C:/Users/Serguei/OneDrive/Documents/Papers/A%20scalable%20method%20for%20time.pdf.

4 Pre-processing by Empirical Mode Decomposition

Empirical Mode Decomposition (EMD), also called a Hilbert-Huang Transform, decomposes a signal into intrinsic mode function. This decomposition is similar to Fourier or Wavelet type transformations however EMD is specifically designed for non-stationary and nonlinear data.

While the application of wavelet transformation is more common in time series literature, some researcher argue that EMD offers better temporal and frequency resolutions..[12] For example, Lin et. al. shows improved foreign exchange forecasts by decomposing the data with EMD before applying ARIMA or SVM models.[13]

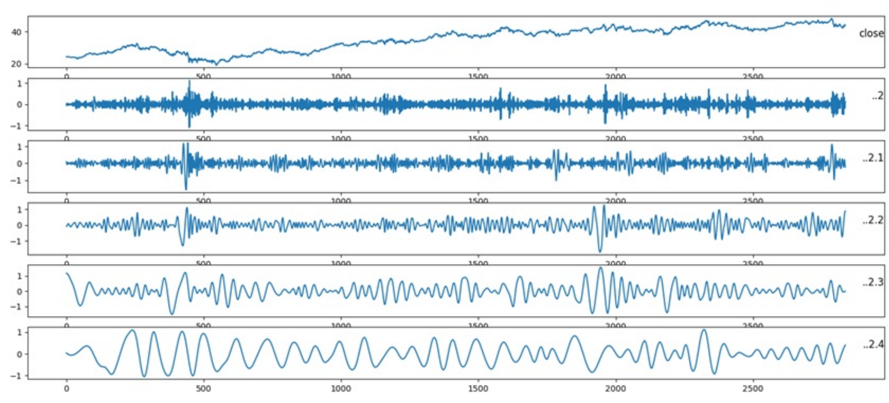

Figure 1: Empirical Mode Decomposition of daily KO stock price data

12. Norden E. Huang et al., "Applications of Hilbert-Huang transform to non-stationary financial time series analysis," *Applied Stochastic Models in Business and Industry* 19, no. 3 (2003): 245–268, ISSN: 1524-1904, doi:10.1002/asmb.501, `http://doi.wiley.com/10.1002/asmb.501`.

13. Chiun Sin Lin, Sheng Hsiung Chiu, and Tzu Yu Lin, "Empirical mode decomposition-based least squares support vector regression for foreign exchange rate forecasting," *Economic Modelling* 29, no. 6 (2012): 2583–2590, ISSN: 02649993, doi:10.1016/j.econmod.2012.07.018, `http://dx.doi.org/10.1016/j.econmod.2012.07.018`.

5 Training

In our research, we haven't found a rule for determining the size of the network. While larger networks are computationally demanding, they didn't lead to overfitting, at least for our task of interest.

As we will present comparison of different RNN architectures later on in the results section, we have observed that GRU cells performed the best for forecasting of stock prices. Since there are too many possible combinations of networks, inputs and other parameters, we have to keep some parts constant. Therefore, unless mentioned otherwise, all the models are composed of GRU cells with 3 layers with 200 units each. Overall, this structure performed better than other recurrent networks and had stable convergence.

For the implementation, we used the Keras package with Tensorflow backend on Python. Keras package offers multiple optimizers for training Recurrent Neural Networks such as RMSprop, SGD (Stochastic Gradient Descent) and Adam. An ideal optimizer is one that converges to an optimal solution without getting stuck in local minima. This requires incorporation of a combination of momentum and decaying learning rate in the optimization algorithm.[14] Adam is one such algorithm and in our experience it produced consistent and stable convergence compared to other choices.

Networks are trained with approximately 9 years long daily price time series and the test set is 140 days long. Inputs are a fixed length univariate or multivariate window of delayed inputs and the target is the one day ahead close price of the stock or index.

---

14. Bianchi et al., "An overview and comparative analysis of Recurrent Neural Networks for Short Term Load Forecasting."

6 Test Results

6.1 Univariate

In this section the inputs to the various RNNs are previous daily close prices only. There are no exogenous variables besides price data following equation 1. We observed that networks composed of GRU cells led to smallest MAPE (Mean Absolute Percent Error) for the test data.

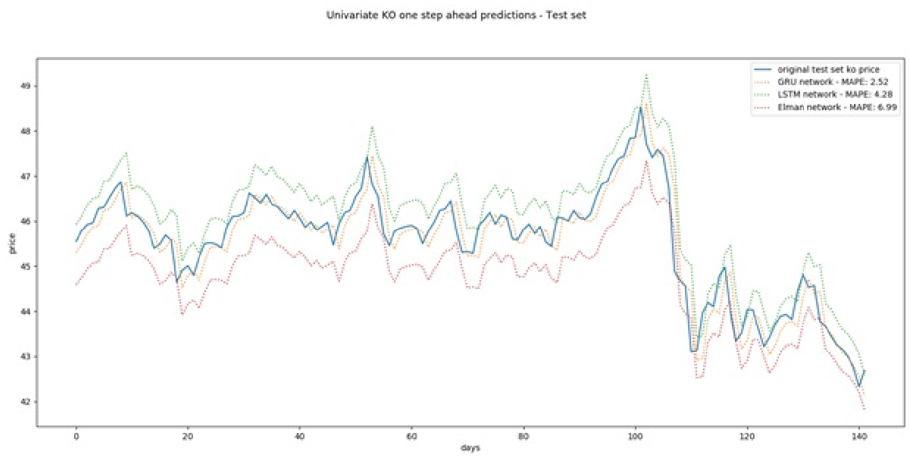

Figure 2: Comparison of forecasting performance of RNN cells on KO stock price

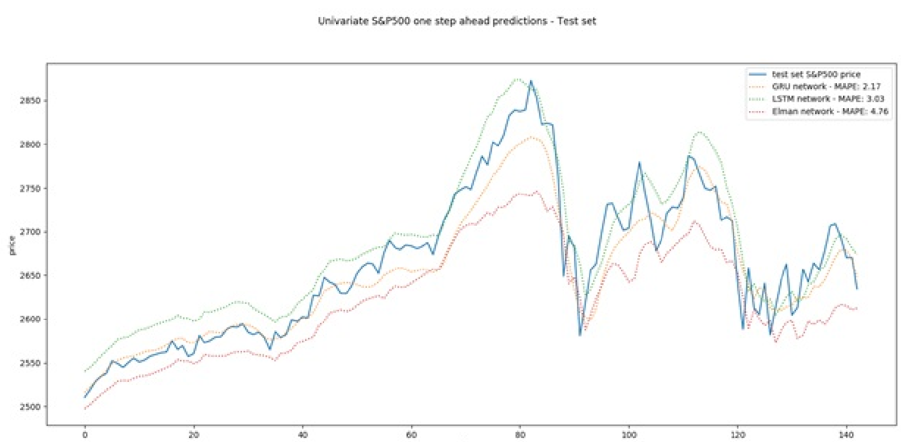

Figure 3: Comparison of forecasting performance of RNN cells on S&P500 index values

6.2 Multivariate

In this section there are lagged exogenous inputs besides the lagged price data. This model follows equation 2.

6.2.1 Empirical Mode Decomposition

Here exogenous inputs besides the price data is the intrinsic mode functions obtained by the Emprical Mode Decomposition.

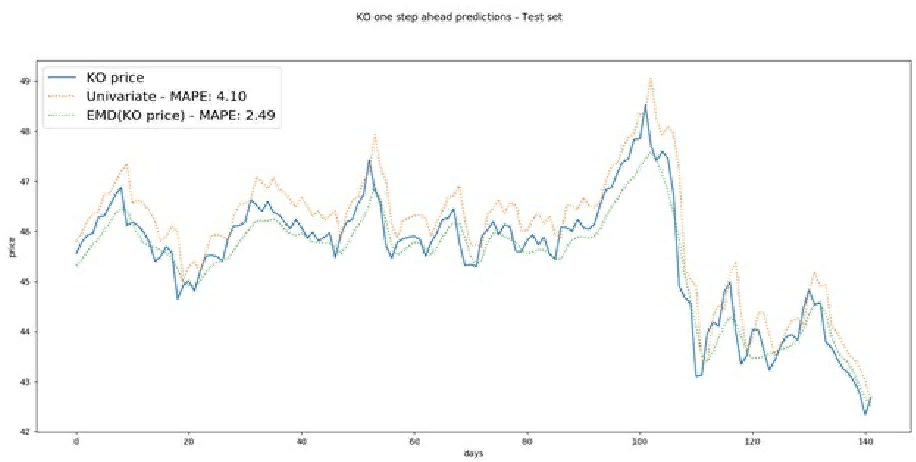

Figure 4: Empirical Mode Decomposition improves KO price forecasts

Figure 5: EMD improves S&P500 index forecasts

6.2.2 Indicator Signals - Clustering by Self Organizing Map

In this section exogenous variable is the mappings of indicator signals into clusters by the SOM algorithm.

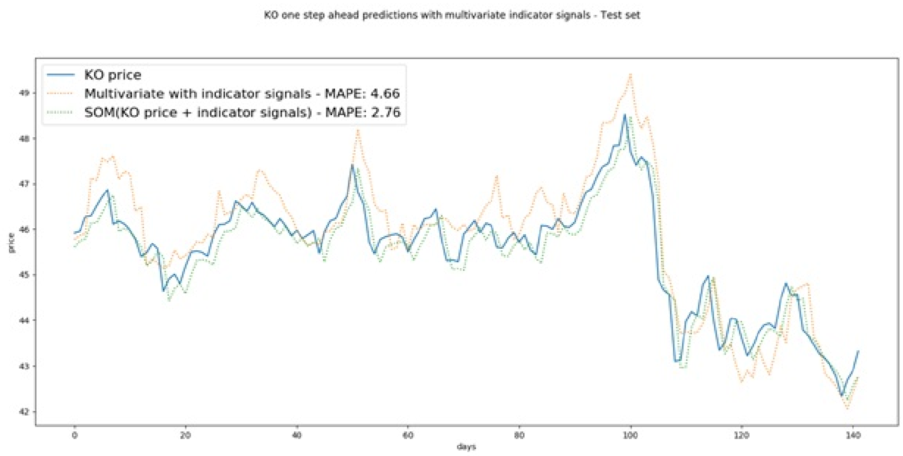

Figure 6: Clustered indicator signals improves forecast errors

7 Conclusions

We have observed that amongst RNNs GRU network perform the best for the task of predicting one step ahead stock prices. Furthermore, we have observed that Empirical Mode Decomposition of the price data or clustering indicator signals with Self Organizing Maps can improve the prediction performance.

While these improvements are relatively significant, forecasting errors still remain large. For KO stock and S&P500 index presented in this paper the lowest one step ahead forecast MAPE remains at approximately 2% and we have observed even larger errors for other stocks. Therefore it is questionable whether this short term forecasting approach can be turned into a profitable trading strategy.

References

Ardalani-Farsa, Muhammad, and Saeed Zolfaghari. "Chaotic time series prediction with residual analysis method using hybrid Elman-NARX neural networks." *Neurocomputing*, 2010. ISSN: 09252312. doi:10.1016/j.neucom.2010.06.004.

Assaad, Mohammad, Romuald Boné, and Hubert Cardot. "LNCS 4233 - Predicting Chaotic Time Series by Boosted Recurrent Neural Networks." *Part II LNCS* 4233 (2006): 831–840.

Bianchi, Filippo Maria, Enrico Maiorino, Michael C Kampffmeyer, Antonello Rizzi, and Robert Jenssen. "An overview and comparative analysis of Recurrent Neural Networks for Short Term Load Forecasting":1–41.

Bonã, Romuald, Michel Crucianu, and Jean-Pierre Asselin De Beauville. "Learning long-term dependencies by the selective addition of time-delayed connections to recurrent neural networks." *Neurocomputing* 48 (2002): 251–266. www.elsevier.com/locate/neucom.

Cherif, Aymen, Hubert Cardot, and Romuald Boné. "SOM time series clustering and prediction with recurrent neural networks." *Neurocomputing*, 2011. ISSN: 09252312. doi:10.1016/j.neucom.2010.11.026.

Hammer, Barbara, Alessio Micheli, Nicolas Neubauer, Alessandro Sperduti, and Marc Strickert. "Self-Organizing Maps for Time Series." *WSOM 5th Workshop On Self-Organizing Maps*, 2005, 1–8. doi:1011619893. http://www.math.unipd.it/~sperduti/PAPERI/wsom2005.pdf%5Cnpapers3://publication/uuid/EB621581-B75E-494C-96B3-7C0C0418DFA4.

Huang, Norden E., Man-Li Wu, Wendong Qu, Steven R. Long, and Samuel S. P. Shen. "Applications of Hilbert-Huang transform to non-stationary financial time series analysis." *Applied Stochastic Models in Business and Industry* 19, no. 3 (2003): 245–268. ISSN: 1524-1904. doi:10.1002/asmb.501. http://doi.wiley.com/10.1002/asmb.501.

Längkvist, Martin, Lars Karlsson, and Amy Loutfi. "A review of unsupervised feature learning and deep learning for time-series modeling." *Pattern Recognition Letters*, 2014. ISSN: 01678655. doi:10.1016/j.patrec.2014.01.008.

Lin, Chiun Sin, Sheng Hsiung Chiu, and Tzu Yu Lin. "Empirical mode decomposition-based least squares support vector regression for foreign exchange rate forecasting." *Economic Modelling* 29, no. 6 (2012): 2583–2590. ISSN: 02649993. doi:10.1016/j.econmod.2012.07.018. http://dx.doi.org/10.1016/j.econmod.2012.07.018.

Marra, Salvatore, and Francesco C Morabito. "A New Technique for Solar Activity Forecasting using Recurrent Elman Networks." *Computational Intelligence* 3, no. 1 (2006): 8–13.

Voegtlin, Thomas. "Recursive self-organizing maps." *Neural Networks* 15, nos. 8-9 (2002): 979–991.

Wang, X, K Smith, R Hyndman, and D Alahakoon. "A scalable method for time series clustering." *Unrefereed research papers* 1 (2004). `papers://b6c7d293-c492-48a4-91d5-8fae456be1fa/Paper/p3095%5Cnfile:///C:/Users/Serguei/OneDrive/Documents/Papers/A%20scalable%20method%20for%20time.pdf`.