Determining the Rolling Window Size of Deep Neural Network Based Models on Time Series Forecasting

Li Shen\textsuperscript{1,2,3}\textsuperscript{*}, Zijin Wei\textsuperscript{2,3}, Yangzhu Wang\textsuperscript{3,4}

\textsuperscript{1,2,3}School of Automation Science and Electrical Engineering, Beihang University, Beijing, China
\textsuperscript{2}School of Computer and Information, Anhui Normal University, Wuhu, Anhui, China
\textsuperscript{*}shenli@buaa.edu.cn, \textsuperscript{3,4}weizijin233@gmail.com, \textsuperscript{4}wangyangzhu@buaa.edu.cn

Abstract: Time series forecasting has always been a significant task in various domains. In this paper, we propose DeepARMA, a LSTM-based recurrent neural network to tackle this problem. DeepARMA is derived from an existing time series forecasting baseline, DeepAR, overcoming two of its weaknesses: (1) rolling window size determination: the way DeepAR determines rolling window size is casual and vulnerable, which may lead to the unnecessary computation and inefficiency of the model; (2) neglect of the noise: pure autoregressive model cannot deal with the condition where data are composed of various kinds of noise, neither do most of time series models including DeepAR. In order to solve these two problems, we first combine a classic information theoretic criterion, AIC, with the network to determine the proper rolling window size. Then, we propose a jointly-learned neural network fusing white Gaussian noise series given by ARIMA models to DeepAR’s input. That is exactly why we name the network ‘DeepARMA’. Our experiments on a real-world dataset demonstrate that our improvement settles those two problems put forward above.

1. Introduction

Time series forecasting has a pivotal role in many application domains, including stock market prediction\cite{1}, demand or supply forecasting\cite{2}, disease propagation analysis\cite{3}, etc. Due to its importance and practicality, a great many different methods have been developed aiming to solve the problem.

Up to now, existing methods related to time series forecasting can be divided into two groups: conventional methods and modern methods. Conventional methods including ARIMA\cite{4}, which is one of the most dominant models, are mainly based on statistics and mathematics. They normally have theoretical guarantees, such as Box-Jenkins methodology\cite{5}, exponential smoothing\cite{6} and state space models\cite{7}. There also exist methods inspired by traditional machine learning algorithms like support-vector machine\cite{8}, hierarchical Bayesian methods\cite{9}. Though conventional methods take advantage of strong interpretability, they suffer from linear assumption, leading to their poor performance in non-linear situations.

Modern methods are mainly based on deep learning and neural networks. Some methods use deep learning approaches to enhance conventional methods. DeepAR\cite{10} builds an LSTM-based\cite{11} auto-regressive RNN to solve the probabilistic forecasting problem. Deep SSM\cite{12} offers the state space model ability to learn complex patterns by combining deep jointly-learned recurrent neural network. In recent years. the introduction of transformer\cite{13} allows the direct pattern extraction from
data with solely DNN. The LogSparse Transformer\[14\] introduces a time series forecasting baseline by applying an improved transformer with LogSparse self-attention to shrink the calculation. To solve long sequence time-series forecasting (LSTF) problems, Informer\[15\] proposes a ProbSparse self-attention mechanism and self-attention distilling operation to reduce the space and time complexity of the model so that the model can deal with longer input.

However, all these models fail to consider the factor of the white noise. Additionally, it has been demonstrated that the rolling window size fundamentally affects the performance of the model. However, rolling window size selection of current models is often casual and unreliable, which may trigger the inefficiency of the model. To solve these problems, we propose DeepARMA, a LSTM-based RNN developed from DeepAR. Our contributions are three fold:

1. We propose a method to determine the rolling window size by combining deep neural network with conventional models to make it interpretable and reliable.
2. DeepARMA takes white Gaussian noise into consideration and treats it as one of the covariates. According to this, DeepARMA performs better than DeepAR especially when time series are mainly composed of noise. In order to acquire white Gaussian noise series, ARIMA is put into use to obtain the variance of the white Gaussian noise.
3. We have successfully applied DeepARMA to time series forecasting with noise and performed experiments on a real-world gyroscope static drift dataset. Results show the validation of the improvement in DeepAR.

2. Methodology

2.1 Problem definition

The problem definition is briefly introduced in this section. Given $N_{t_0}$-length time series $\{z_{i,1:t}^N\}_{i=1}^N$ and associated $(t_0+T)$-length covariates $\{x_{i,1:t_0+T}^N\}_{i=1}^N$, our goal is to predict the corresponding unknown $T$-length time series $\{z_{i,t_0+1:t_0+T}^N\}_{i=1}^N$. $z_{i,t_1:t_2}$ here represents the value of time series $i$ during time index $t_1$ to time index $t_2$ and $x_{i,t_1:t_2}$ represents covariates that are somehow associated to $z_{i,t_1:t_2}$ and also assumed to be known all the time. Furthermore, both $z_{i,t_1:t_2}$ and $x_{i,t_1:t_2}$ can be univariate or multivariate. In this way, we are able to model the following conditional distribution.

$$P(z_{i,t_0+1:t_0+T} \mid z_{i,1:t_0}, x_{i,1:t_0+T})$$

(1)

Note that $t_0$ and $T$ are learnable parameters respectfully standing for the rolling window size and predicting steps.

2.2 DeepAR

Our model, DeepARMA, is developed from an existing autoregressive LSTM-based network——DeepAR\[10\]. We emphasize that we are not claiming to invent a new autoregressive recurrent architecture but simply propose a network derived from DeepAR which achieves better generalization performance. We will briefly introduce its core architecture and refer readers to [10] for more details.

DeepAR assumes that the model distribution $Q_\theta(z_{i,t_0+1:t_0+T} \mid z_{i,1:t_0}, x_{i,1:t_0+T})$ is composed of the multiplication of likelihood factors

$$\prod_{t=t_0+1}^{t_0+T} Q_\theta(z_{i,t} \mid z_{i,1:t_0}, x_{i,1:t_0+T}) = \prod_{t=t_0+1}^{t_0+T} l(h_{i,t} \mid \theta(h_{i,t}, \Theta))$$

(2)

parametrized by the output $h_{i,t}$ of an autoregressive LSTM-based recurrent neural network. The likelihood $l$ should be a fixed distribution and chosen related to the statistical properties of the time series.
2.3 Determining the proper window size

The sliding windows size determines the complexity of the model to a large extent especially during training the network. Little attention had been paid to it including DeepAR. In our approach, we apply AIC\[^{16}\], a classic information theoretic criterion and an estimate of a measure of fit of the model, to the determination of the proper window size. AIC is defined by

\[
AIC = (-2) \log(l) + 2k
\]

where \( l \) refers to the maximum likelihood and \( k \) refers to the number of independently adjusted parameters within the model. However, it is still a tough work to estimate the model with every possible window size in the limited range. We address this issue by firstly using ARIMA to make the prediction and computing the corresponding smallest AIC\((p, q)\) which stands for the most appropriate value of AIC for the model with AR order \( p \) and MA order \( q \). On this foundation, the rolling window size could be chosen due to the maximum of \( p \) and \( q \). Despite ARIMA is a conventional model based on linear assumption, it can at least give the minimum of the proper window size. In our study, we will take twice the maximum of \( p \) and \( q \) merging with the prediction length as the default rolling window size.

2.4 Replacing AR with ARMA

In most of prevalent modern time series forecasting models, no matter which kind of architecture they are based on, CNN, RNN or Transformer, hardly ever do they consider the effect of the white Gaussian noise. It matters a lot when the input of the time series are mainly composed of combination of the white Gaussian noise where MA model is more likely to match. Therefore, instead of building an autoregressive model like DeepAR, we contribute DeepARMA, an autoregressive moving average LSTM-based recurrent neural network.

As is shown in Figure 1, the network receives the inputs, including the previous values of the time series \( z \), covariates \( x \) and white Gaussian noise \( \xi \) when the other parts mirror the architecture of DeepAR. Notice that the variance of the white Gaussian noise \( \xi \) is given by the ARIMA\((p, q)\) model which is also used to determine the rolling window size.

3. Results

3.1 Datasets and environment

We perform experiments on a real-world dataset. This gyroscope static drift dataset consists of 57936 groups of data collected from a static silicon micro-gyroscope and accelerometer. Each group of data includes recorded time, 3-axis acceleration and 3-axis angular velocity. It should be mentioned that these data ought to be recorded at a fixed interval of 5 ms. However, it might slightly change from time to time due to the restrictions of experiment equipment. We evaluate our model only using...
z-axis’s sensor data owing to the assumption that 3-axis sensor data are independent from each other. Notice that these sensor data are supposed to be mainly composed of various kinds of noise.

We have selected two time-series forecasting methods as comparison, including ARIMA and DeepAR. We implement our model using Pytorch and all the models were trained/tested on a single Nvidia GeForce GTX 1050. Both models contain a 3-LSTM-layer stack and are optimized with Adam optimizer whose learning rate is 1e-6. The total number of epochs is 40 for both and the batch size is chosen as 32.

3.2 Experiments

We first show the procedure to determine the window size and how effective it is to choose a proper window size. Above all, we use ARIMA model to make the prediction in order to find out the corresponding smallest AIC(p, q). The result of the calculation of AIC(p, q) is shown in Table 1.

| p  | AIC | q  |
|----|-----|----|
| 0  | -7.102 | 0  |
| 1  | -7.478 | -7.495 |
| 2  | -7.500 | -7.520 |
| 3  | -7.506 | -7.520 |
| 4  | -7.536 | -7.540 |
| 5  | -7.561 | -7.655 |
| 6  | -7.733 | -7.834 |
| 7  | -7.812 | -7.921 |

As is shown in Table1, AIC(2, 6) has the smallest value, implying that the rolling window size should be over than 6. Thus, assuming that the prediction length is also twice the maximum of p and q, we determine the default rolling windows size as 24.

Further, we compare with other two baselines in the condition of different rolling windows size, including the default one. We consider ND (Normalized Deviation) as the criterion to judge the accuracy, which is defined as

\[
ND = \frac{1}{\sum_{i,t} |z_{i,t} - \hat{z}_{i,t}|} \sum_{i,t} |z_{i,t} - \hat{z}_{i,t}|
\]

\(\hat{z}_{i,t}\) refers to the predicted median value of variable \(i\) at time \(t\). The results are summarized in Table 2.

| Rolling Window Size | Baselines | DeepARMa | DeepAR | ARIMA(2,6) |
|---------------------|-----------|----------|--------|------------|
| 16                  | 0.300     | 13min20s | 0.403  | 12min34s   | 0.527       |
| 18                  | 0.355     | 16min40s | 0.291  | 14min37s   |             |
| 24                  | 0.154     | 19min6s  | 0.174  | 15min51s   |             |
| 30                  | 0.188     | 27min57s | 0.254  | 17min43s   |             |
| 36                  | 0.187     | 32min30s | 0.273  | 21min10s   |             |
| 48                  | 0.175     | 39min15s | 0.485  | 27min48s   |             |
3.3 Discussions
Observed from Table 2, DeepARMA outperforms the other baselines and the chosen default window size obviously performs best compared with other longer and shorter ones. This illustrates that taking white Gaussian noise into consideration, DeepARMA is a more competitive model especially when the data is mainly composed of noise. It is also obvious that the rolling window determination method is effective.

4. Conclusions
In this paper, we improve DeepAR to a more advanced model, DeepARMA. Our experiments on gyroscope static drift dataset suggests that DeepARMA can effectively learn a global model from time series mainly composed of noise while DeepAR may suffer. We also showed that rolling window size plays a key role on the complexity and efficiency of time series forecasting models. Our proposed method to determine the rolling window size had been examined to be valid. Exploring better strategy in rolling window size determination and extending our method to fit other time series forecasting baselines are our future research directions.

Acknowledgements
This work was supplied and supported by the National Key Research and Development Program of China(NO. 2017YFB0503002).

Reference
[1] Kimoto T., Asakawa K., Yoda M., et al.(1990) Stock market prediction system with modular neural network. In: International Joint Conference on Neural Networks.
[2] Helmick J. S. (1999) Designing and managing the supply chain: concepts, strategies, and case studies. Journal of Business Logistics, 22(1):259-261.
[3] Matsubara Y., Sakurai Y, Panhuis W, et al. (2014) FUNNEL: automatic mining of spatially coevolving epidemics. In: ACM.
[4] George E. B. and Gwilym M. J. (1968) Some recent advances in forecasting and control. Journal of the Royal Statistical Society. Series C (Applied Statistics), 17(2):91–109.
[5] George E. B., Gwilym M. J., Gregory C. R., and Greta M. L. Time series analysis: forecasting and control. Journal of Time, 31(3).
[6] Hyndman R., Koehler A.B., Ord J.K., and Snyder R.D. (2008)Forecasting with Exponential Smoothing: The State Space Approach. Springer Series in Statistics.
[7] Durbin J. (2012)Time series analysis by state space methods. Oxford University Press, Oxford.
[8] Kim K.j. (2003)Financial time series forecasting using support vector machines. Neurocomputing, 55: 307-319
[9] Chapados N. (2014)Effective bayesian modeling of groups of related count time series. arXiv preprint arXiv:1405.3738.
[10] Flunkert V., Salinas D., and Gasthaus J.(2017)Deepar: Probabilistic forecasting with autoregressive recurrent networks. International Journal of forecasting, 36(3): 1181-1191
[11] Hochreiter S., Schmidhuber J.(1997) Long Short-Term Memory. Neural Computation, 9(8):1735-1780.
[12] Rangapuram S. S., Seeger M., Gasthaus J., Stella L., et al. (2018) Deep state space models for time series forecasting. In: International Conference on Neural Information Processing Systems. Curran Associates Inc., Red Hook, NY, USA, pp.7796–7805.
[13] Vaswani A., Shazeer N.M., Parmar, N., et al.(2017). Attention is All you Need. ArXiv, abs/1706.03762.
[14] Li S., Jin X., Xuan Y., et al.(2019) Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting. arXiv:1907.00235.
[15] Zhou H., Zhang S., Peng J., et al. (2020)Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting . arXiv:2012.07436.
[16] Akaike H. T. (1974) A new look at the statistical model identification. IEEE Transactions on Automatic Control, 19(6):716-723.