ARMemNet: Autoregressive Memory Networks for Multivariate Time Series Forecasting

Jinuk Park
Yonsei University
Republic of Korea
parkju536@yonsei.ac.kr

Chanhee Park
Yonsei University
Republic of Korea
channy_12@yonsei.ac.kr

Hongchan Roh
SK Telecom
Republic of Korea
hongchan.roh@sk.com

Sanghyun Park*
Yonsei University
Republic of Korea
sanghyun@yonsei.ac.kr

ABSTRACT

Recently, several studies show the powerful capability of neural networks to capture non-linear features from time series which have multiple seasonal patterns. However, existing methods rely on convolution kernels implicitly, hence neglect to capture strong long-term patterns and lack interpretability. In this paper, we propose a memory-augmented neural network named AutoRegressive Memory Network (ARMemNet) for multivariate time series forecasting. ARMemNet utilizes memory components to explicitly encode intense long-term patterns. Furthermore, each encoder is designed to leverage inherently essential autoregressive property to represent short-term patterns. In experiments on real-world dataset, ARMemNet outperforms existing baselines and validates effectiveness of memory components for complex seasonality which is prevalent in time series datasets.

CCS CONCEPTS

• **Mathematics of computing → Time series analysis**

KEYWORDS

Multivariate Time Series Forecasting, Memory Augmented Neural Networks, Multiple Seasonal Patterns

1 INTRODUCTION

Forecasting time series has a fundamental problem in many applications, such as environmental system [10], financial market [13], and industrial purpose. In such modern systems, modeling and forecasting time series is crucial to make a decision for the future, to predict trends, or to detect abnormal events. Without external information, forecasting time series is concluded by identifying the seasonality and modeling dependencies between time steps and variables. Therefore, the major challenge in forecasting time series is how to derive significant seasonality in the past patterns and to attend those periodicities in future prediction.

Unfortunately, complex seasonality which involves more than one repeating pattern is prevalent in real-world time series. Multiple patterns are generally divided into two categories; short-term patterns that is repeated patterns in short time intervals, and long-term patterns that lie in relatively long intervals. For instance, a hourly electricity consumption time series can have distinct patterns for weekday and weekend corresponding to the long-term patterns which can be explained by high power demand for industrial usage during the weekday, while there is no such demand on the weekend. On the other hand, the short-term patterns can be defined as the repeating electricity consumption pattern within a day (e.g., the lowest demand in dawn or highest peak around at noon) [7]. Since those long-term and short-term patterns are interrelated, it is non-trivial to capture complex patterns if there are multiple seasonal components in time series.

Several methods have been proposed to forecast time series based on the autoregressive property of time series [5, 9]. Especially, recurrent neural networks (RNN) and its variants [2, 3] have been widely used in time, however, they suffer from the long-term dependency problem. Recently, convolutional neural networks (CNN) [13] coupled with attention mechanism have achieved prominence in time series analysis. However, they are limited to implicitly encode temporal patterns using kernels. This implicit extraction of seasonality roughly captures the complex seasonal patterns of the time series, so that it results in poor prediction and interpretability.

To address these limitations, we present a novel method named AutoRegressive Memory Network (ARMemNet) for efficient time series forecasting when the series contains multiple seasonal patterns. In most of time series, especially those from industrial purposes, some seasonal patterns are conspicuous enough to be easily identified. Focusing on this observation, we utilize memory components to model long-term seasonal patterns explicitly. We adopt attention mechanism for aligning information in encoded current short-term and each memory component to forecast future value. Furthermore, we propose a novel non-linear version of autoregressive encoder to capture short-term patterns while retaining the power of autoregressive property. Additionally, we incorporate a linear autoregressive model to alleviate the scale insensibility.

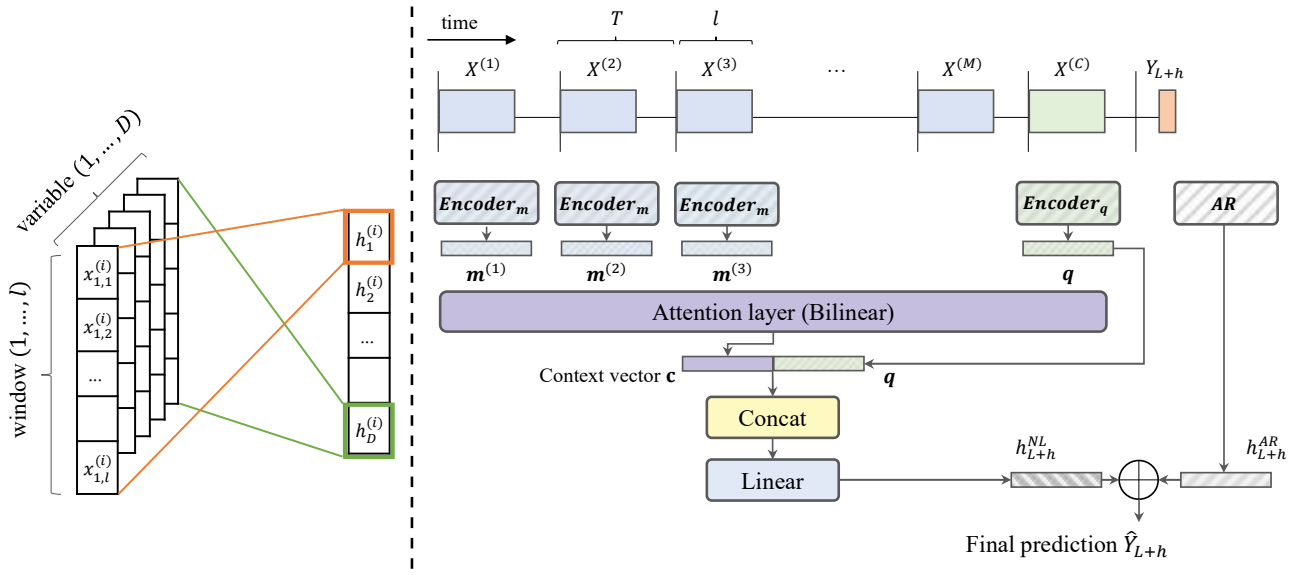
The main contributions of this paper are as follows:

- We introduce a memory-augmented neural network with autoregressive encoders to model complex seasonal patterns.

* **Corresponding Author**: sanghyun@yonsei.ac.kr

SAC'21, March 22 –March 26, 2021, Virtual Event, South Korea

DOI: https://doi.org/10.1145/3412841.3442108



(a) Encoder Architecture (b) An Overview of ARMemNet

Figure 1: (a) Autoregressive encoder design. (b) An overview of the autoregressive memory network (ARMemNet)

We explicitly encode seasonality using memory components and simplify encoders for the long-and short-term.

- In several experiments on real-world datasets, the proposed model outperforms the existing approaches when the time series have distinct or even indistinct patterns.
- We show that the proposed memory component and attention mechanism efficiently capture complex patterns.

2 RELATED WORK

One of the traditional off-the-self time series analysis tools is the autoregressive integrated moving average (ARIMA) [1]. ARIMA models and their variants are often favorable due to the interpretability. Other machine learning techniques including vector autoregression (VAR) [9], linear support vector regression (LSVR) [14], regularized linear regression [12], and Gaussian Process [11] have the advantages of being simple and well-interpreted. However, they need distributional assumptions and short ability to model non-linearity.

Recently, deep neural networks have been brought tremendous attention to the powerful ability to capture non-linearity in time series. Especially, LSTNet [7] leverages both the convolutional and recurrent layer to model time series. However, it fails to work on dynamic patterns. Focusing on this issue, DSANet [4] was proposed to model dynamic-period patterns. It uses convolutional layers with self-attention for both local and global patterns. However, it has high computation cost for stacked self-attention layer since it takes a multivariate time series into several univariate series and computes each series.

3 METHODS

Figure 1 shows an overview of ARMemNet architecture. ARMemNet consists of three major components: two encoders for memories and short-term, attention layer, and autoregressive module. We start with the problem formulation for time series forecasting and then describe each component in the following part.

Problem Formulation Consider a fully observed multivariate time series $Y = \{y_1, y_2, \dots, y_L\}$ where $y_t \in \mathbb{R}^D$ and D is the variable dimension. The goal of one-step time series forecasting is to predict h ahead y_{L+h}. Since we set a one-step prediction problem, to predict y_{L+h+1}, we assume $\{y_1, y_2, \dots, y_L, y_{L+1}\}$ is observed. We do not consider any exogenous information.

To define input series X, we subset historical memories and the current short-term series from $\{y_t\}_{t=1}^{L}$. We divide the whole series into $M+1$ sub-series with the length of T each. Then we define the first M sub-series as M memories $(X^{(1)}, X^{(2)}, \dots, X^{(M)})$ and the last sub-series as the current short-term input $(X^{(C)})$. The unit length T and the number of memories M are tunable hyperparameters. We denote the i-th memory as $X^{(i)} \in \mathbb{R}^{D \times T}$, and a univariate time series for the variable j in i-th memory as $X_j^{(i)} = \left\{x_{j,1}^{(i)}, x_{j,2}^{(i)}, \dots, x_{j,T}^{(i)}\right\}$. For the experiments, we use l time stamps (windows) among the length T for each sub-series. Figure 1 (b) illustrates the formulation of the memories and the current series.

Encoder Architecture We introduce a non-linear version of autoregressive encoder (AR encoder, Figure 1 (a)). We focus on encoding a univariate time series while retaining powerful autoregressive property. The dimension of AR encoder output vector is equal to the variable dimension D since it computes AR projection for each variable. Given input $X^{(i)}$ matrix, the output representation is denoted as $h^{(i)} = \left\{h_1^{(i)}, h_2^{(i)}, \dots, h_D^{(i)}\right\} \in \mathbb{R}^D$. If we denote by $h_j^{(i)}$ the output representation for the variable j in

i-th memory, the operation of $AREncoder$ for variable j can be written as:

$$h_j^{(i)} = \sigma\left(\sum_{t=1}^{l} W_{j,t} x_{j,t}^{(i)} + b_j\right) \quad (1)$$

where σ denotes non-linear activation function, $W_j \in \mathbb{R}^l$ and b_j denote a learnable parameter and bias for variable j, respectively. As described in the equation, the representation is encoded by its past observations for each variable j. This is the reason why we named it as an autoregressive encoder. Also, we employ one shared encoder for all long-term historical memories to generate explicitly encoded memories $m^{(i)} \in \mathbb{R}^D$ and another encoder for the current query $q \in \mathbb{R}^D$.

$$m^{(i)} = AREncoder_m\left(X^{(i)}\right) \quad (2)$$

$$q = AREncoder_q\left(X^{(C)}\right) \quad (3)$$

where $i = 1, \dots, M$ and $AREncoder$ denotes the autoregressive encoders.

Attention Layer To predict the complex periodicity which changes dynamically over time, we adopt attention mechanisms between encoded memories $\{m^{(i)}\}_{i=1}^{M}$ and query q. We employ bilinear function [8] to compute attention scores. The attention weights $\{\alpha_i\}_{i=1}^{M}$ for memories and a context vector $c \in \mathbb{R}^D$ are computed as follows:

$$\alpha_i = \mathrm{softmax}\left(q^{\mathrm{T}} W_s m^{(i)}\right) \quad (4)$$

$$c = \sum_{i=1}^{M} \alpha_i m^{(i)} \quad (5)$$

where $\mathrm{softmax}(X_i) = \exp(X_i)/\sum_i \exp(X_i)$ and $W_s \in \mathbb{R}^{D \times D}$ is a learnable parameter to allow flexibility in the attention scores.

We obtain final input representation by concatenating the weighted context vector c and the current query q, and feed it into a linear transformation for the non-linear part of prediction h_{L+h}^{NL}.

$$h_{L+h}^{NL} = W_c[q; c] + b_c \quad (6)$$

Autoregressive Component Since the proposed model is based on non-linearity, the scale of the predicted often does not properly reflect the scale of inputs. To mitigate this problem, we fuse a linear autoregression to the model. The prediction of linear autoregression for variable j can be formulated as:

$$h_{j,L+h}^{AR} = W_j^{AR}\left[X^{(1)}; X^{(2)}; \dots; X^{(M)}; X^{(C)}\right] + b_j^{AR} \quad (7)$$

where $W_j^{AR} \in \mathbb{R}^{l \times (M+1)}$ and b_j^{AR} are learnable parameters for the variable $j = 1, \dots, D$ and l denotes window size of inputs.

Finally, we obtain the final prediction $\hat{Y}_{L+h}$ of ARMemNet by integrating non-linear and linear prediction.

$$\hat{Y}_{L+h} = h_{L+h}^{NL} + h_{L+h}^{AR} \quad (8)$$

The whole training scheme is performed by stochastic gradient descent using Adam optimizer [6] with L2 regularized MAE loss.

4 EVALUATION

Datasets We evaluated the proposed method on four public datasets. For all datasets, we used 60% to train, 20% to validate, and 20% to test in time order. For reproducing, we used the preprocessed datasets from [7].

- Solar-Energy: The solar-energy dataset comprises of solar power production records in every 10 minutes from 137 plants in Alabama state, 2006. The number of variables D is 137.
- Traffic: The traffic dataset consists of 48 months hourly road occupancy rate at 862 sensors ($D = 862$) from the California Department of Transportation (2015-2016).
- Electricity: The electricity consumption datasets are observed from 321 clients ($D = 321$) (2012-2014).
- Exchange-Rate: The dataset contains daily exchange rates for Australia, British, Canada, Switzerland, China, Japan, New Zealand and Singapore (1990-2016) ($D = 8$).

Baselines We evaluate our model compared to seven baseline methods. For traditional baselines, we choose VAR [1], LRidge, LSVR [14], and GP [11]. We also compare to three neural methods including LSTNet [7] and DSANet [4].

We used root relative squared error (RRSE) and empirical correlation coefficient (CORR) for the evaluation metrics. A prediction is better for lower RRSE and higher CORR.

Experimental Details Following previous studies, we conducted a grid search in similar range for all possible tunable hyperparameters. For DSANet [4], we set the same search spaces from the original paper. For ARMemNet, we set $M = 7$ for all datasets except for the Exchange-Rate. Since the Exchange-Rate is daily time series, we set M is chosen from $\{1, 2\}$ and $T = 5$. Also, we choose the unit length of memory $T = 144$ for Solar-Energy, 24 for Electricity and Traffic datasets. The window size l is chosen from $\{2^0, 2^1, \dots, 2^7\}$. To be specific, we limited the maximum search space of l not to over the unit length T in each dataset, respectively.

Main Results Table 1 reports the main results of all methods on the test set. Note that since we used the same test set, we brought the reported scores from the original LSTNet paper [7] except for DSANet.

In the result table, ARMemNet outperforms the baseline methods in 14 of 16 test cases based on RRSE. This indicates that memory components are helpful for capturing and forecasting time series when it has multiple seasonal patterns. Especially, since Exchange-Rate time series has dynamic patterns rather than fixed seasonal terms, it is much difficult to forecast the future. However, due to the flexible attention mechanism with encoded memories, ARMemNet results in the best performance.

Effect of Memory Components To further investigate the effect of memory components, we visualize each memory and attention scores with corresponding short-term series. We plot randomly selected variables at random target timestamp offset in Electricity dataset. Note that red lines denote the memories used for $X^{(i)}$ and green line denotes the short-term input $X^{(C)}$. The attention score for corresponding each memory appears above the red line.

Table 1: Evaluation results of all methods on four datasets. The best performance is highlighted in bold in each case.

Datasets		Solar-Energy				Traffic				Electricity				Exchange-Rate			
		Horizon				Horizon				Horizon				Horizon			
Methods	Metrics	3	6	12	24	3	6	12	24	3	6	12	24	3	6	12	24
AR	RSE	0.2435	0.3790	0.5911	0.8699	0.5991	0.6218	0.6252	0.6293	0.0995	0.1035	0.1050	0.1054	0.0228	0.0279	0.0353	0.0445
	CORR	0.9710	0.9263	0.8107	0.5314	0.7752	0.7568	0.7544	0.7519	0.8845	0.8632	0.8591	0.8595	0.9734	0.9656	0.9526	0.9357
LRidge	RSE	0.2019	0.2954	0.4832	0.7287	0.5833	0.5920	0.6148	0.6025	0.1467	0.1419	0.2129	0.1280	**0.0184**	0.0274	0.0419	0.0675
	CORR	0.9807	0.9568	0.8765	0.6803	0.8038	0.8051	0.7879	0.7862	0.8890	0.8594	0.8003	0.8806	**0.9788**	**0.9722**	0.9543	0.9305
LSVR	RSE	0.2021	0.2999	0.4846	0.7300	0.5740	0.6580	0.7714	0.5909	0.1523	0.1372	0.1333	0.1180	0.0189	0.0284	0.0425	0.0662
	CORR	0.9807	0.9562	0.8764	0.6789	0.7993	0.7267	0.6711	0.7850	0.8888	0.8861	0.8961	0.8891	0.9782	0.9697	0.9546	0.9370
GP	RSE	0.2259	0.3286	0.5200	0.7973	0.6082	0.6772	0.6406	0.5995	0.1500	0.1907	0.1621	0.1273	0.0239	0.0272	0.0394	0.0580
	CORR	0.9751	0.9448	0.8518	0.5971	0.7831	0.7406	0.7671	0.7909	0.8670	0.8334	0.8394	0.8818	0.8713	0.8193	0.8484	0.8278
GRU	RSE	0.1932	0.2628	0.4163	0.4852	0.5358	0.5522	0.5562	0.5633	0.1102	0.1144	0.1183	0.1295	0.0192	0.0264	0.0408	0.0626
	CORR	0.9823	0.9675	0.9150	0.8823	0.8511	0.8405	0.8345	0.5300	0.8597	0.8623	0.8472	0.8651	0.9786	0.9712	0.9531	0.9223
LSTNet-skip	RSE	0.1843	0.2559	0.3254	0.4643	0.4777	0.4893	0.4950	0.4973	0.0864	0.0931	0.1007	0.1007	0.0226	0.0280	0.0356	0.0449
	CORR	0.9843	0.9690	0.9467	0.8870	0.8721	0.8690	0.8614	0.8588	0.9283	0.9135	0.9077	0.9119	0.9735	0.9658	0.9511	0.9354
LSTNet-attn	RSE	0.1816	0.2538	0.3466	**0.4403**	0.4897	0.4973	0.5173	0.5300	0.0868	0.0953	0.0984	0.1059	0.0276	0.0321	0.0448	0.0590
	CORR	0.9848	0.9696	0.9397	**0.8995**	0.8704	0.8669	0.8540	0.8429	0.9243	0.9095	0.9030	0.9025	0.9717	0.9656	0.9499	0.9339
DSANet	RSE	0.2182	0.2740	0.4538	0.5457	0.5702	0.5899	0.5911	0.5442	0.0881	0.1013	0.1074	0.1068	0.0234	0.0287	0.0348	0.0448
	CORR	0.8835	0.9639	0.8963	0.8300	0.8004	0.7845	0.7831	0.8160	0.9285	0.9056	0.8805	0.8765	0.9727	0.9652	0.9535	0.9344
AR-MemNet	RSE	**0.1813**	**0.2381**	**0.3217**	0.4460	**0.4604**	**0.4725**	**0.4732**	**0.4838**	**0.0820**	**0.0918**	**0.0970**	**0.0986**	0.0188	**0.0244**	**0.0235**	**0.0253**
	CORR	**0.9848**	**0.9724**	**0.9472**	0.8950	**0.8785**	**0.8714**	**0.8655**	**0.8601**	**0.9475**	**0.9357**	**0.9278**	**0.9225**	0.9776	0.9691	**0.9697**	**0.9680**

As shown in Figure 2, the model assigns the higher attention scores for the relevant memories based on the query; hence the model can obtain higher accuracy in the experiments. This observation strongly indicates that the attention mechanism in the proposed model can effectively capture the useful information and empower the forecasting performance.

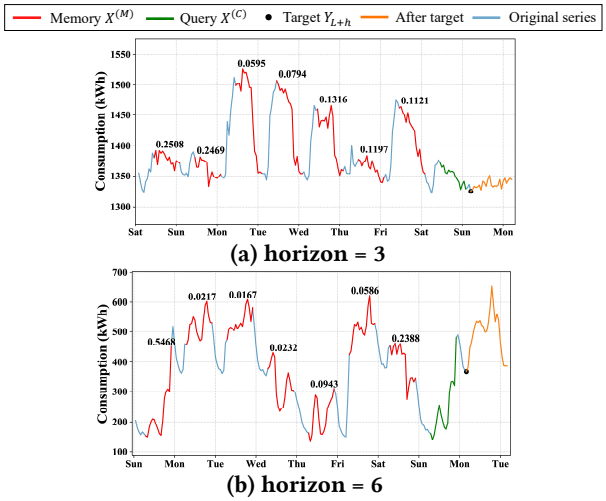


Figure 2: Effect of memory components in Electricity.

5 CONCLUSION

In this paper, we present a novel method, autoregressive memory networks (ARMemNet). ARMemNet utilizes memory components to explicitly encode strong patterns with autoregressive encoders and employs attention mechanism to integrate them. The experiments on real-world datasets demonstrates the strong performance of ARMemNet.

ACKNOWLEDGMENTS

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (IITP-2017–0–00477)

REFERENCES

[1] Box, G.E.P., Jenkins, G.M., Reinsel, G.C. and Ljung, G.M. 2015. *Time series analysis: forecasting and control.* John Wiley & Sons.
[2] Chung, J., Gulcehre, C., Cho, K. and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555.* (2014).
[3] Hochreiter, S. and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Computation.* 9, 8 (Nov. 1997), 1735–1780. DOI:https://doi.org/10.1162/neco.1997.9.8.1735.
[4] Huang, S., Wu, X., Wang, D. and Tang, A. 2019. DSANet: Dual self-attention network for multivariate time series forecasting. *International Conference on Information and Knowledge Management, Proceedings* (2019).
[5] Hyndman, R.J. and Athanasopoulos, G. 2018. *Forecasting: principles and practice.* OTexts.
[6] Kingma, D.P. and Ba, J.L. 2015. Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings* (2015).
[7] Lai, G., Chang, W.C., Yang, Y. and Liu, H. 2018. Modeling long- and short-term temporal patterns with deep neural networks. *41st International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2018* (2018).
[8] Luong, M.T., Pham, H. and Manning, C.D. 2015. Effective approaches to attention-based neural machine translation. *Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing* (2015).
[9] Melnyk, I. and Banerjee, A. 2016. Estimating structured vector autoregressive models. *33rd International Conference on Machine Learning, ICML 2016* (2016).
[10] Pang, Y., Yao, B., Zhou, X., Zhang, Y., Xu, Y. and Tan, Z. 2018. Hierarchical electricity time series forecasting for integrating consumption patterns analysis and aggregation consistency. *IJCAI International Joint Conference on Artificial Intelligence* (2018).
[11] Roberts, S., Osborne, M., Ebden, M., Reece, S., Gibson, N. and Aigrain, S. 2013. Gaussian processes for time-series modelling. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences.* (2013). DOI:https://doi.org/10.1098/rsta.2011.0550.
[12] Saleh, A.K.M.E., Arashi, M. and Kibria, B.M.G. 2019. *Theory of Ridge Regression Estimation with Applications.* John Wiley & Sons.
[13] Tsantekidis, A., Passalis, N., Tefas, A., Kanniainen, J., Gabbouj, M. and Iosifidis, A. 2017. Forecasting stock prices from the limit order book using convolutional neural networks. *Proceedings - 2017 IEEE 19th Conference on Business Informatics, CBI 2017* (2017).
[14] Vapnik, V., Golowich, S.E. and Smola, A. 1997. Support vector method for function approximation, regression estimation, and signal processing. *Advances in Neural Information Processing Systems* (1997).