ATPAM: Adversarial Temporal Pattern Attention Mechanism

Luyao Huang

School of Southwest Jiao Tong University, Chengdu, China

282967361@qq.com

Abstract. Time series forecasting has a large number of applications in daily life, for instance the prediction of stock prices, electricity consumption, exchange rate changes etc. However, the existing time series prediction methods have limitations. The most significant one is that when all the prediction models get the predicted value, a new round of iteration starts after the loss function is calculated with the corresponding real data. This may cause the accumulation of errors, since there is only one loss function to measure the difference between the predicted value and the ground truth, it will make the connection weak and mainly depend on the accuracy of the prediction model. Unfortunately, there are few time series prediction models with high accuracy in reality. To solve the issue, in this paper, we propose a new time series forecasting model – Adversarial Temporal Pattern Attention Mechanism (ATPAM), which based on Generative Adversarial Nets (GANs). ATPAM adopts a Temporal Pattern Attention model as the generator to learn time-invariant temporal patterns, and use a discriminator to improve the prediction performance and do auxiliary adversarial training. Extensive experiments on several real-world datasets show the effectiveness of our method.

Keywords: time series forecasting, neural networks, generative adversarial networks, temporal pattern attention.

1. Introduction

Time series forecasting (TSF) models are widely used in business, finance and engineering. Lots of classical approaches are proposed to solve these time series forecasting problems, the two main types are statistics and deep learning. Autoregressive Integrated Moving Average [1] etc. is the first type which has well performance in linear time series prediction. However, when the dimension of time series data increases and the relationship between features becomes complicated, the model has a bottleneck. So, deep learning network like Long short-term Memory (LSTM) [2], Gate Recurrent Unit (GRU) [3] etc. are used frequently in most specific time series forecasting problems.

In recent years, with the improvement of computer hardware resources and the popularization of neural networks, more and more time series forecasting models are built with deep neural networks. [4] [5,6] mainly use Recurrent Neural Network (RNN) to achieve TSF, others like [7-10] have using attention mechanism to do. However, all these models get the final results by optimizing only one objective function, like MSE loss. So, this may degrade the performance of the predictive model. Therefore, we should appropriately increase number of objective function while selecting a suitable one to achieve better prediction results.

In 2014, Generative Adversarial Networks (GANs) [11] were proposed and first applied in the field of image processing. This is a landmark research achievement by having neural networks play against with each other rather than human intervention. Motivated by GANs, TPA-LSTM [10] and [6], in this paper, we propose Adversarial Temporal Patterns Attention (ATPAM). It combines the ideas of both GANs and TPA-LSTM.

The main contribution of our paper are as follow:

We propose an effective time series forecasting model – Adversarial Temporal Patterns Attention based on Generative Adversarial Nets and Temporal Patterns Attention. Extensive experiments on different real-world time series datasets show the effectiveness of our model, moreover, auxiliary adversarial training improves the robustness and generalization of model. To capture both past and future features in the time series sequence, we replace regular long-and short term memory network (LSTM) to bidirectional LSTM (BiLSTM). BiLSTM combine two hidden states, and has a better performance than LSTM. To extract features in hidden matrix step by step, we use dilation convolution to achieve it. In fact, adopting the dilation rate can slow down feature loss, and improve forecasting accuracy. The rest of this paper is organized as follows. Section 2 gives a related work about our proposed model. Section 3 presents the problem formulation and background. Section 4 introduces the framework of our model. Section 5 aims at demonstrating the effectiveness of our methods on real-word time series datasets. Finally, we draw a conclusion in Section 6.

2. Related work

Time Series Forecasting (TSF) task is one of the most significant branches of data mining, The classical TSF model is autoregressive(AR) which proposed by British statistician G.U.Yule [12]. The output of AR model are linearly dependent on their previous values and random bias, other AR model like moving average (MA) model [13] and autoregressive moving average model are proposed successively. However, all above models lack ability to deal with non-stationary time series data. Due to this reason, the Autoregressive Integrated Moving Average model (ARIMA) [1] is proposed later. Then, Thissen, UVBR and Van Brakel et al. and Gui, Bin and Wei et al. put forward a new model which based on support vector machine (SVM), it also can handle TSF problems called support vector regression (SVR). SVR maps time series data from original space to high dimensional space. Although it has a better performance than ARIMA, we expect the model to be far more accurate than that.

Recent years, deep neural network have proposed for TSF. Model based on deep neural network has great advantages in solving nonlinear problems. To alleviate this issue, Lai, Guokun and Chang et al. [5] give a Long and Short-Term Temporal Patterns model (LSTNet). they propose a novel recurrent-skip component which leverages the periodic pattern in real-word sets. Nonetheless, LSTNet has some major shortcomings:

- First is that p is an empirical parameter, it is not universal in specific tasks.
- Next, due to p, this model is more likely designed for periodic data while in real life, not all time series data are periodic.
- Last, LSTNet selects a relevant hidden state as in typical attention mechanism.

Based on LSTNet, Shih, Shun-Yao and Sun et al. [6] remove the hyperparameter p and make the entire model more suitable for general time series data.

Since Generative Adversarial Net (GAN) was proposed by Goodfellow, Ian and Pouget-Abadie et al. [11], it first shines in the field of image generation. This is a landmark research achievement by having neural networks play against with each other rather than human intervention. After that, various variants of GAN have emerged and so on. Although GANs have powerful generation capabilities, the application field of GANs is limited to the image field, and it is unable to do anything in time series prediction until the emergence of C-RNN-GAN . C-RNN-GAN applies the GANs architecture to generate sequential melody data. Yoon, Jinsung and Jarrett et al. first utilize GANs to generate time series which called timeGAN. However, these time series data are only generated to get as close as possible to the history data, and cannot forecast future steps. Motivated by them, we improve the TPA-LSTM model, change its LSTM layers to GRU layers, and then, attach a discriminator to the output of our proposed model, it becomes a TSF model with adversarial training. Experiments show that our model has better performance than original TPA-LSTM and timeGAN.

3. Background

3.1. Problem definition

Suppose that we have a task of Multivariate Time Series Forecasting(MTSF), the input of MTSF is $X = \{x_1, x_2, \cdots, x_{t-1}\}$, where $x_i \in R^n$ represents the observed values at time i, we are going to predict the values of $x_{t-1+\Delta}$, where Δ is a appropriate horizon in different situations. At the same time, we let $y_{t-1+\Delta}$ be the ground truth, and let $\hat{y}_{t-1+\Delta}$ be the predicted values, which means $\hat{y}_{t-1+\Delta} = x_{t-1+\Delta}$. Besides, we predict the next one step by the first $(t - w)$ steps, where w is a window size [5].

3.2. Temporal Patterns Attention

We refer to TPA-LSTM by taking advantage of Temporal Patterns Attention mechanism. To capture both long-term and short-term dependencies in the time series sequence, Shih et al. and Lai, Guokun et al. utilize a Long short-term memory (LSTM) to handle long-term dependencies in sequence which defined as following:

$$I_t = \sigma(X_t W_{xi} + H_{t-1} W_{hi} + b_i) \tag{1}$$

$$F_t = \sigma\left(X_t W_{xf} + H_{t-1} W_{hf} + b_f\right) \tag{2}$$

$$O_t = \sigma(X_t W_{xo} + H_{t-1} W_{ho} + b_o) \tag{3}$$

$$\tilde{C}_t = \tanh(X_t W_{xc} + H_{t-1} W_{hc} + b_c) \tag{4}$$

$$C_t = F_t \odot C_{t-1} + I_t \odot \tilde{C}_t \tag{5}$$

$$H_t = O_t \odot \tanh(C_t) \tag{6}$$

where W_{xi}, W_{xf} , W_{xo} , $W_{xc} \in R^{d\times h}$ denote input weight parameters, b_i, b_f, b_o , $b_c \in R^{1\times h}$ denote bias parameters, W_{hi}, W_{hf} , W_{ho} , $W_{hc} \in R^{h\times h}$denote hidden state parameters, C_t, $\tilde{C}_t$ and $H_t \in R^{n\times h}$, σ denotes activate function, and $\odot$ denotes element-wise multiplication.

Next, to capture short-term dependencies and enhance the learning ability of the model, Shih et al. use 2-D CNN to convolve the hidden state of the sequence, which we show in Figure 1 and define as following:

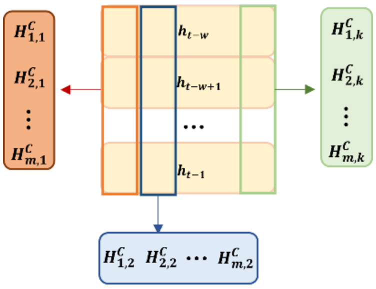

Figure 1. Convolution to extract features from hidden states

$$H^C_{i,j} = \sum_{l=1}^{w} H_{i,(t-w-1+l)} \times C_{j,T-w+l} \tag{7}$$

where $H^C_{i,j}$ is the convolved values, T is the maximum length we intend to observe, w is window size, C_i is filters.

Last but not the least, using Temporal Patterns Attention mechanism to get the final output. Here is a brief introduction shown in Figure 2, and refer to [6] to get more details:

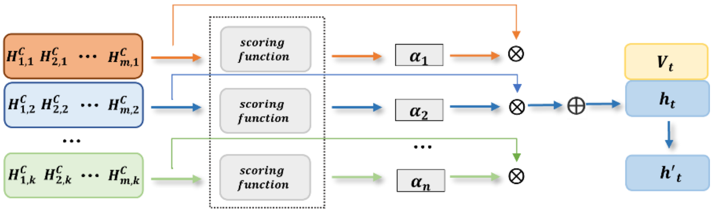
$H^C_{1,1}$ $H^C_{2,1}$ ··· $H^C_{m,1}$
$H^C_{1,2}$ $H^C_{2,2}$ ··· $H^C_{m,2}$
$H^C_{1,k}$ $H^C_{2,k}$ ··· $H^C_{m,k}$
scoring function
scoring function
scoring function
α_1
α_2
α_n
V_t
h_t
h'_t

Figure 2. Prediction process in Temporal Pattern Attention.

$$\alpha_i = \sigma\left(\left(H_i^C\right)^T W_\alpha h_t\right) \tag{8}$$

$$v_t = \sum_{i=1}^{n} \alpha_i H_i^C \tag{9}$$

$$h^{'}_t = W_h h_t + W_v v_t \tag{10}$$

$$\hat{y}_{t-1+\Delta} = W_{h^{'}} h^{'}_t \tag{11}$$

where H_i^C is the ith row of matrix H^C,$W_\alpha \epsilon R^{k\times m}$, $h_t \epsilon R^{m\times 1}$, h_t,$h^{'}_t \in R^m$,$W_h \in R^{m\times m}$,$W_v \in R^{m\times k}$, and $W_{h^{'}} \in R^{n\times m}$ and $\hat{y}_{t-1+\Delta} \in R^n$.

Equation 8 to 11, we utilize convolved matrix from hidden states and obtain attention weight α by scoring function. After getting context vector, we integrate the output and last time step hidden state to get predicted values.

4. Framework of ATPAM

4.1. Model Architecture

We first elaborate on the general framework of our model. As illustrated in Figure 3, The proposed model is a Temporal Patterns Attention with adversarial training. The adversaries are two different recurrent neural models, a generator (G) and a discriminator(D). We consider the entire Temporal Patterns Attention as G. After getting the predicted value by G, we compare it with the real data to get the loss of the generator. Then, we utilize the D to distinguish the similarity between the generated fake data and the real data, getting the discriminator loss, that is, adversarial loss. The target of G is to generate fake data that is indistinguishable from real data, while D needs to accurately discriminate the fake data generated by G. The process of adversarial training becomes a zero-sum game until both G and D reach the Nash equilibrium. In other words, G produces data that infinitely close to the real data, and D cannot distinguish between the data generated by G and the real data, it tends to be a random judgment. We define the following loss function L_D and L_G :

$$L_G = \frac{1}{m}\sum_{i=1}^{m} \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right) \tag{12}$$

$$L_D = \frac{1}{m}\sum_{i=1}^{m}\left[-\log D\left(x^{(i)}\right) - \left(\log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right)\right)\right] \tag{13}$$

where $z^{(i)}$ is training data, and $x^{(i)}$ is real data, m is the num of samples.

4.2. Dilation Convolution on Hidden States

In Section 4.1, we have an overview of Adversarial Temporal Pattern Attention. Through Gated Recurrent Units (GRU), we get all t sequence step hidden states from h_1 to h_t. In TPA-LSTM, model utilizes convolution to extract features from matrix $Hidden$(mentioned in Section 3), which directly change channels from $hiddenuntis$ to 1. This strategy can extract features rapidly, however, each sliding window compress all features into 1 dimension will sacrifice the accuracy of the representation. In our model, we modify it to dilation convolution, by controlling dilation rate, we extract the features of the latent variable matrix multiple times within a sliding window which illustrated in Figure 3. Due to dilation convolution, we can learn better feature representations in the same sliding window, which means that we can better capture the feature information of past moments, improve the accuracy of prediction, and increase the prediction effect of the model.

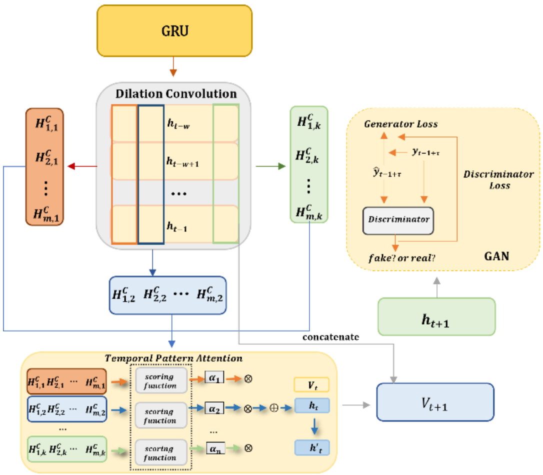

Figure 3. The framework of Adversarial Temporal Patterns Attention. h_t represents the hidden state of the Bidirectional Long-and short term Memory Network at time step t. We use 2-D CNN filters to convolve features of hidden states, and get a matrix H^C. Next, using Attention mechanism calculates each row of H^C to generate V_t. Then, we concatenate V_t and h_t to generate $\hat{y}_{t-1+\tau}$ as output of generator. Finally, we get generator loss by utilized L1Loss and calculate adversarial loss by using BCELoss. This adversarial training becomes a zero-sum game until both Generator and Discriminator reach the Nash equilibrium.

4.3. Discriminator

As illustrated in Figure 3, the discriminator consists of a two layer bidirectional recurrent network, three groups of linear dense and activate function [10]. In the model, the recurrent network utilized is the Long short-term Memory Network [2] and groups of linear dense add nonlinear fitting capability of the model. The output of the generator is fed into the discriminator together with the ground truth, while passing through the bidirectional LSTM layers, the discriminator take context in both directions into account for its decisions. The discriminator consists of tow parts, the first one is that distinguish fake data generated by G to False, and the second one is that the real data needs to be discriminated as True. To implement this simple binary classification problem, we choose the Binary Cross-Entropy Loss (BCELoss) as adversarial function, defined as following:

$$\hat{L}_{D1} = \frac{1}{m}\sum_{i=1}^{m} -\lambda \cdot \left[Z_{real}. \log D\left(x^{(i)}\right) + (1 - Z_{real}) \cdot \log\left(1 - x^{(i)}\right)\right] \tag{14}$$

$$\hat{L}_{D2} = \frac{1}{m}\sum_{i=1}^{m} -\lambda \cdot \left[Z_{fake}. \log D\left(G\left(z^{(i)}\right)\right) + \left(1 - Z_{fake}\right) \cdot \log\left(1 - G\left(z^{(i)}\right)\right)\right] \tag{15}$$

$$\hat{L}_D = \hat{L}_{D1} + \hat{L}_{D2} \tag{16}$$

where $x^{(i)}$is real data, $z^{(i)}$is training data, $\boldsymbol{Z_{real}}$ is a vector of all ones with the same shape as $D\left(x^{(i)}\right)$, $\boldsymbol{Z_{fake}}$ is a vector of all zeros with the same shape as $G\left(z^{(i)}\right)$. λ is a hyperparameter, for single-label binary classification tasks, it does not matter to set λ or not.Finally, we add $\hat{L}_{DI}$ and $\hat{L}_{D2}$ to get the complete objective function of the discriminator, as shown in eq.5, and our goal is to minimize this equation.

5. Experimental studies

In this experimental section, we first introduce the datasets which used. Next, we show other baseline methods compared with our model. Then, we give the metrics to evaluate the model. Last but not least, we describe some experiment details. Finally, we discuss the ablation study on our model.

5.1. Data

We use four publicly available benchmark datasets Exchange Rate: The exchange rates for eight countries from 1990 to 2016 (Australia, British, Canada, China, Japan, New Zealand, Singapore, and Switzerland).

5.2. Baseline Methods for Comparison

We compare our proposed model with several baseline models on typical MTS datasets:

- TPA-LSTM: Temporal Patterns Attention with LSTM layer, an improvement on the LSTNet model.
- ATPAM: our proposed model base on TPA-LSTM with GRU layer and adversarial training.

5.3. Metrics

We use two traditional evaluation metrics defined as following:

- Root Relative Squared Error (RSE):

$$RSE = \frac{\sqrt{\sum_{it\epsilon\Omega_{valid}}(y_{it} - \hat{y}_{it})^2}}{\sqrt{\sum_{it\epsilon\Omega_{valid}}(y_{it} - \bar{y})^2}} \tag{17}$$

- Relative Absolute Error (RAE):

$$RAE = \frac{\left|\sum_{it\epsilon\Omega_{valid}}(y_{it} - \hat{y}_{it})\right|}{\left|\sum_{it\epsilon\Omega_{valid}}(y_{it} - \bar{y})\right|} \tag{18}$$

where Ω_{valid} is validation set in the source datasets, $\bar{y}$ is mean of Ω_{valid} , and $\hat{y}_{it}$ is predictive values, y_{it} is ground truth in Ω_{valid} . In order to understand which model perform best in experiments, we observe and record the values of RAE and RSE. From equation 15 and 16, we know that, if predicted value is near to the real data, in other words, $y_{it} - \hat{y}_{it}$ is closer to zero, RAE and RSE will be smaller, and the model have a better performance.

5.4. Results

Due to the improvement on Temporal Pattern Attention with dilation, our model is better at extracting features from the time series in long intervals. Moreover, we find that when the horizon intervals becomes very long, we need a larger batch size to update, such as 256, 512 or larger. However, when the horizon is small, we only need to update through each tiny batch size. And the loss value in training stage shown in Figure 4.

Table 1. Forecasting result of ATPAM.

Methods	Metrics	horizon			
		48	72	96	120
TPA-LSTM	RAE	0.110	0.120	0.132	0.138
TPA-LSTM	RSE	0.112	0.122	0.135	0.140
ATPAM	RAE	**0.078**	**0.098**	**0.111**	**0.124**
ATPAM	RSE	**0.095**	**0.114**	**0.130**	**0.136**

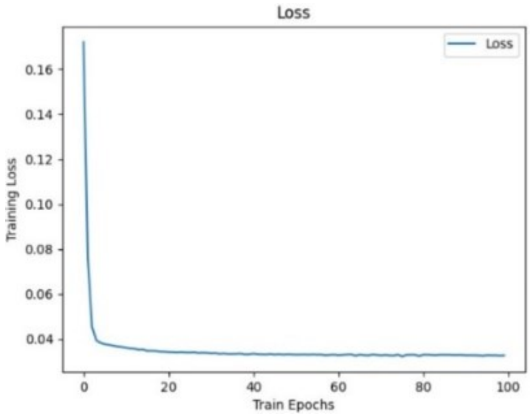

Figure 4. Training Loss under different epochs on exchange rate with ATPAM.

6. Conclusions

In this paper, we present ATPAM, a model based on TPA-LSTM with replace LSTM layer to GRU layer and attach adversarial training. Extensive experiments in Section 5 on several real-world datasets show the effectiveness of our method. However, due to the large gap in structural complexity between the generator and the discriminator, it is hard for the G and the D to converge at the same time. Moreover, improved model involves two neural network, which has high time complexity.

Next, we will simplify the structure of the generator, and pay more attention to the data preprocess.

References

[1] David J Bartholomew. Time series analysis forecasting and control., 1971.

[2] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.

[3] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches.arXiv preprint arXiv:1409.1259, 2014.

[4] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. International Journal of Forecasting, 36(3):1181–1191, 2020.

[5] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pages 95–104, 2018.

[6] Shun-Yao Shih, Fan-Keng Sun, and Hung-yi Lee. Temporal pattern attention for multivariate time series forecasting. Machine Learning, 108(8):1421–1441, 2019.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. Advances in neural information processing systems, 30, 2017.

[8] Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. Advances in Neural Information Processing Systems, 32, 2019.

[9] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In Proceedings of AAAI, 2021.

[10] Sifan Wu, Xi Xiao, Qianggang Ding, Peilin Zhao, Ying Wei, and Junzhou Huang. Adversarial sparse transformer for time series forecasting. Advances in Neural Information Processing Systems, 33:17105–17115, 2020.

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. Advances in neural information processing systems, 27, 2014.

[12] Jeffrey R Russell and Robert F Engle. A discrete-state continuous-time model of financial transactions prices and times: The autoregressive conditional multinomial–autoregressive conditional duration model. Journal of Business & Economic Statistics, 23(2):166–180, 2005.

[13] GI Taylor. Proceedings of the royal society of london. series a, containing papers of a mathematical and physical character. The formation of emulsions in definable fields of flow, pages 501–523, 1934.