Temporal Convolutional Networks for Anomaly Detection in Time Series

IOP Conf. Series: Journal of Physics: Conf. Series **1213** (2019) 042050 doi:10.1088/1742-6596/1213/4/042050

Temporal Convolutional Networks for Anomaly Detection in Time Series

Yangdong He and Jiabao Zhao

School of Management & Engineering, Nanjing University, Nanjing 210093, China

jbzhao@nju.edu.cn

Abstract. Convolutional Networks have been demonstrated to be particularly useful for extracting high level feature in structural data. Temporal convolutional network (TCN) is a framework which employs casual convolutions and dilations so that it is adaptive for sequential data with its temporality and large receptive fields. In this paper, we apply TCN for anomaly detection in time series. We train the TCN on normal sequences and use it to predict trend in a number of time steps. Prediction errors are fitted by a multivariate Gaussian distribution and used to calculate the anomaly scores of points. In addition, a multi-scale feature mixture method is raised to promote performance. The validity of this method is confirmed on three real-world datasets.

1. Introduction

Nowadays there are large numbers of sequences of multivariate time series data produced in a wide range of scenes. Anomaly detection in time series means to find the potential patterns in series data that significantly differ from normal behavior. It has attracted significant intersect in community for its applications in real-world engineering problem.

For the past few years, deep learning has been one of the most popular methods of machine learning, for its outstanding results in a widely range of tasks. One main reason for the success of deep learning is its ability to extract high-level feature automatically from structural data [1]. For sequential data, it is meaningful to extract the inherent patterns of the series data for analyzing, and deep learning is an effective method for its powerful representation capability. Convolutional networks are naturally used to predict future elements of a sequence by slide a 1-D convolutional filter over the data [2]. Recurrent neural networks (RNN) are widely used in sequential task for its ability to store past information in time series. Long short term memory (LSTM) neural networks [3] and Gated Recurrent Unit (GRU) [4] overcome the vanishing gradient trouble in RNN and are applied successfully in speech and natural language processing.

Considering solving anomaly detection problems using networks, because the quantities of normal and anomaly instances are extremely imbalanced, one method to detection anomaly in time series is to train a network for prediction to model normal behavior, and use the errors of prediction to recognize anomalous behavior. It is based on the intuition that after a predictor network has been trained on normal behaviors well, the prediction results in region with anomalous behavior will deviate from the truths for its lack of comprehension on anomalous patterns.

In this paper, the time series anomaly detection problem is solved in an unsupervised learning way, where a temporal convolutional network framework [5] is used as a predictor model and use multivariate

Gaussian distribution to identify anomaly points in time series. Owing to casual convolutions and dilations, TCN is suitable for modeling sequential data for its temporality and flexible receptive fields. Skipping connection is raised to implement multi-scale feature mixture prediction for using varying scale patterns.

2. Convolutional Sequence Model

Given a time series X, where each point $x_t \in R^m$ in time series is a vector, which represents the input variables at time step *t*. A prediction model aims to predict the next *l* values for taking a window of length *L* over the past time series values as input variables. After training the Temporal Convolution Networks based prediction model on a normal time series, the residuals between the prediction values and the real values are calculated, then fit a multivariate Gaussian distribution model. The prediction errors are then used to calculate the probability of the points in the test time series being anomalous.

2.1. Temporal Convolutional Networks

A temporal convolutional network is trained to predict the next *l* values of input time series. Suppose that a sequence of inputs $x_0, x_1, ..., x_L$ is given, and is wished to predict some corresponding output $y_0, y_1, ..., y_L$ at each time step, whose values are equal to the inputs shifted forward *l* time steps. The main constraint is that when predicting the output y_t for some time step *t*, it can only use the inputs that have been observed previously: $x_0, x_1, ..., x_t$.

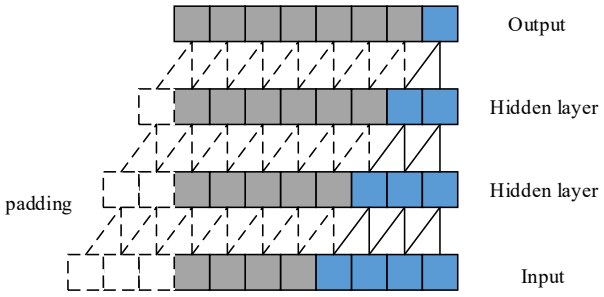


Figure 1. A casual convolution with filter kernel size k=2

The TCN has two main constraints: the output of the network should have the same length as its input, and network can only use the information from past time steps. To fulfil these temporality principles, a 1-D fully-convolutional network architecture [6] is used in TCN, for its all convolution layers have the same length, with zero padding to ensure higher layers the same length as previous ones. Besides, TCN uses causal convolutions, which in each layer an output at time step *t* is computed only with the region no later than time step *t* in the previous layer, as shown in Fig.1. For 1-D data, the causal convolution can be easily implemented by shifting the output of a normal convolution by a few time steps.

2.2. Dilated Convolutions

Generally, when dealing with a time series, it is expected that the networks can memorize long term information. However, with the sample causal convolution we showed previously, the receptive field sizes are limited unless stacking lots of layers. This makes problems when applying the casual convolution on sequence task for its heavy calculating cost. To overcome the problems, dilated convolutions [7] are employed to enable an exponentially large receptive field with limit layers.

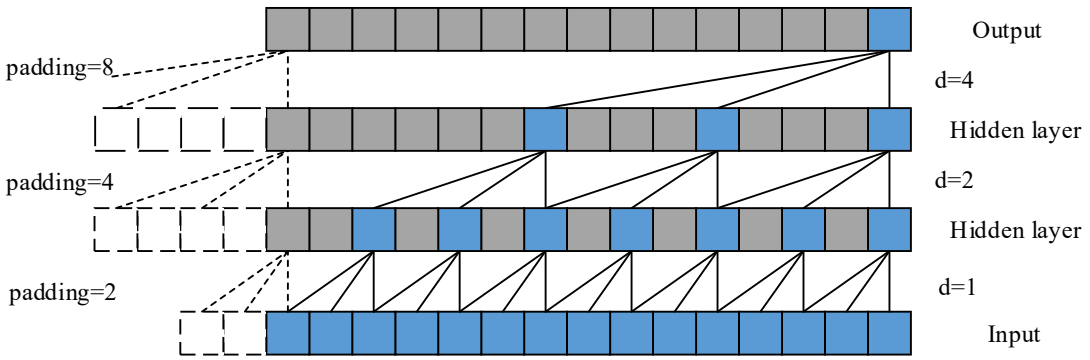


Figure 2. A dilated casual convolution with dilated factors d=1, 2, 4 and filter size k=3.

A dilated convolution is a convolution where filter is applied over a region larger than its size by skipping input values with a given step. This is similar to pooling or stride convolutions for it enhances receptive field size, however it makes the output size equal to the input.

When using dilated convolutions, it is common to increase the dilated factor d exponentially with the depth of the network. This ensures the receptive field covering each input in the history, and enabled to get an extremely large receptive field as effective history by using deep networks. A dilated casual convolution is showed in Fig.2.

2.3. Residual Connections

For the receptive field size of the TCN depends on the network depth n as well as filter size k and dilation factor d, making the TCN deeper and larger is important to obtain large enough receptive field. Empirically, making network deep and narrow, which means stacking a large amount layers and choosing a thin filter size, is an effective architecture.

Residual connections [8] have proven to be very effective in training deep networks. In a residual network, skip connections are used throughout the network, to speed up training process and avoid vanishing gradient problem in deep models.

2.4. Multi-scale Feature Maps for Prediction

Time series prediction model is designed to learn the inherent pattern from past input. Usually, a time series contains different scale patterns, e.g. a power demand data presents daily repetition as well as weekly repetition. Intuitively, the feature layers in different depth in convolution networks with different receptive field can focus on learning different scale patterns.

Inspired by multi-scale object detection networks [9], rather than using only the last layer, it would be better to extract multiple layers feature and combine them together then to make prediction. For the convenient of that each feature map contains the same length in all layers, such feature combining can be easily accomplished via skip connections, as showed in Fig 3. Branches are added from different layers and concatenate them together to get a layer which contains multi-scale feature, and use a 1×1 convolution over this layer to make prediction.

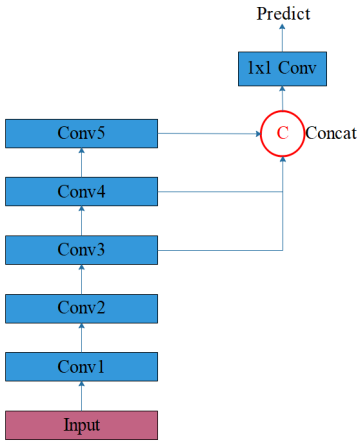


Figure 3. A network using multi-scale feature to predict

3. Anomaly Detection

With the prediction errors as indicators for anomaly, anomaly detection is implemented in point-wise. Prediction errors are the residuals of the predicted values and the truth values. The prediction errors distribution on training data is modeled using a multivariate Gaussian distribution, where the mean and the variance of the Gaussian distribution from error vectors are calculated by using maximum likelihood estimation (MLE).

For getting anomaly score to describe the probability of a point be anomalous in test, the anomaly score of an observation prediction error vector e_t is defined as follow,

$$a_t = (e_t - \mu)^T \Sigma^{-1} (e_t - \mu),$$

where μ and Σ are the mean and co-variance of the prediction errors distribution, and e_t is the prediction error vector in test. A point $\mathrm{x}^{(t)}$ is classified as anomalous if $a_t > \tau$, otherwise is classified as normal. To determine the value of τ, the anomaly score is calculated on the validation set, and a series of logarithmic spacing threshold values are set between zero and maximum of anomaly score. The threshold that maximizes the F_β-score as τ is chosen, where F_β-score is the weighted harmonic mean of precision and recall.

set V_1 for early stop, the validation set V_2 contains anomalies and the test set T with both normal and anomalies. The anomaly detection algorithm is showed as follows:

1. Train the prediction model on training set N, and use V_1 to prevent the model overfitting on the training data by using early stop method.
2. Fit a multivariate Gaussian distribution by the prediction errors on training set N
3. Apply the trained prediction model on V_2 and calculate the anomaly scores by the parameters in the Gaussian distribution. Set the threshold τ which maximizes the F_β-score.
4. Evaluate the prediction model and threshold τ on test set T, and get the results.

4. Experiments

The TCN are tested on different real-world datasets: ECG, space shuttle and 2-D gesture. For each dataset the anomaly is labeled with window-wise, i.e. if a window contains an anomalous pattern, the entire window is labeled as anomalous. Before input the data into networks, each channel of data is normalized to avoid the interference of varying scale data.

In our experiments, the results with and without multi-scale feature to predict are compared. We use the same network and use 1×1 convolution for last layer to get prediction results. For skipping connection to get multi-scale feature map, the last 3 layers are concatenated as we presented in section 2.4. Adam Optimizer [10] is used for minimizing MSE between truth values and predictions. The dataset in [11] is used as it provided labeled anomalies. Table 1 show the results.

Table 1. Precision, Recall and F_β-score for TCN with and without using multi-scale feature. TCN-ms means TCN using multi-scale feature.

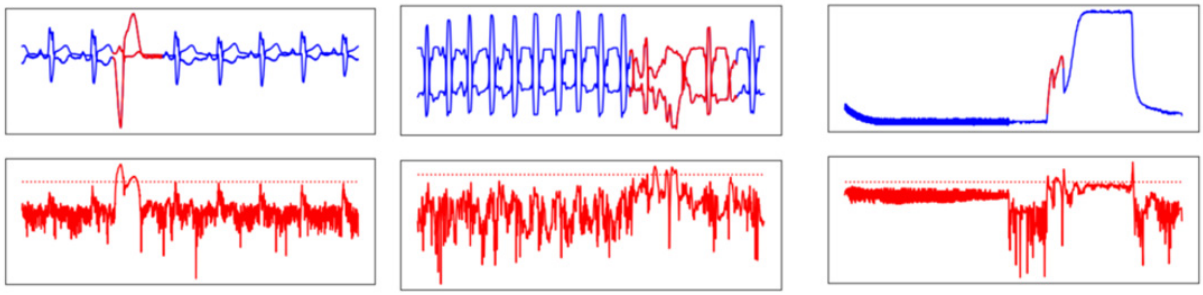
Dataset		ECG	Space shuttle	gesture
β		0.1	0.05	0.05
Precision	TCN	0.930	0.853	0.769
	TCN-ms	0.946	0.880	0.800
Recall	TCN	0.264	0.203	0.126
	TCN-ms	0.299	0.226	0.103
F_β-score	TCN	0.901	0.846	0.759
	TCN-ms	0.908	0.874	0.787

4.1. Datasets

Electrocardiograms (ECG) are time series recording the electrical activity of the heart. There are two channels in the ECGs series data and each of the two traces are searched independently, which contains one anomaly (see fig 4(a). The both channels are used and set window size L=300 to overlap an entire repeating cycle.

2-D Gesture dataset is extracted from a video of an actor performing various actions with and without a replica gun. The two channels time series measure the X and Y coordinates of the actor's right hand. There are a few frames the actor is laughing and flailing hand so it can be recognized as anomalous (see fig 4(b)).

Space shuttle dataset has both short time-period patterns and long time-period patterns. The anomalous cycle has a double hump (see fig 4(c)).



(a) ECG (b) gesture (c) space shuttle

Figure 4. Sample original anomalous sequences (first row) with corresponding anomaly scores (second row). The red regions in the original time series are labeled anomalous and the dashed lines denote the thresholds. The anomaly scores are on log –scale.

4.2. Results

The main observations from our experiments are as follows:

(1) In figure 4 the anomaly scores in the anomalous regions are clearly higher than that in the normal regions for all dataset. However, we do not get anomaly scores over the threshold all along the anomalous regions. In fact, there are many normal points belong to normal behavior in the anomalous window. And that explains the results as table 1 shows that recall is notably lower than precision. A tiny β in F_β-score is used to eliminate the inherent influence of 'anomalous' labeled subsequences and pay more attention to precision.

(2) The results in table 1 show that networks architecture with multi-scale features perform better than original one with higher precision and F_β-score, which concludes that our skip connection for multi-scale feature mixture is effective. In fact, the modification in networks is tiny and simple but it gives good rewards.

(3) We experiment with setting varying window lengths, and for each window length we tune different layers count and filter size to guarantee that receptive field covers the window length. For time series with periodism like ECG and gesture dataset, we set window length equal to and larger than the cycle length, and both of them get good performances. For long time-period data such as space shuttle dataset, we limit window length less than a cycle length. It still performs well in anomalous region while gets a false positive instance in normal subsequence for its lack of information.

5. Conclusion

We show that temporal convolutional networks are able to learn inherent patterns in sequential data automatically and so TCN could be a feasible model to learn normal time series behaviors, and can be used for anomaly detection. Our TCN-based approach works well on three open real-world datasets. Besides, with skip connection for multi-scale feature mixture it can get greater performance at a small cost.

References

[1] Deng L , Yu D . Deep learning: methods and applications[J]. Foundations & Trends in Signal Processing, 2014, 7(3):197-387.
[2] Lecun Y , Bengio Y . Convolutional networks for images, speech, and time series[M]. The handbook of brain theory and neural networks. MIT Press, 1998.
[3] Graves A. Long Short-Term Memory[M]// Supervised Sequence Labelling with Recurrent Neural Networks. 2012.
[4] Chung J , Gulcehre C , Cho K H , et al. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling[J]. Eprint Arxiv, 2014.
[5] Oord A V D , Dieleman S , Zen H , et al. WaveNet: A Generative Model for Raw Audio[J]. 2016.
[6] Long J , Shelhamer E , Darrell T . Fully Convolutional Networks for Semantic Segmentation[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2014, 39(4):640-651.
[7] Yu F , Koltun V . Multi-Scale Context Aggregation by Dilated Convolutions[J]. 2015.
[8] He K , Zhang X , Ren S , et al. Deep Residual Learning for Image Recognition[J]. 2015.
[9] Hariharan B , Arbelaez P , Girshick R , et al. Object Instance Segmentation and Fine-Grained Localization Using Hypercolumns[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017, 39(4):627-639.
[10] Kingma D P , Ba J . Adam: A Method for Stochastic Optimization[J]. Computer Science, 2014.
[11] Keogh E , Lin J , Fu A . HOT SAX: Efficiently Finding the Most Unusual Time Series Subsequence[C]// IEEE International Conference on Data Mining. IEEE, 2006.