Time Series Classification Using Images

Władysław Homenda
the Faculty of Mathematics and Information Science
the Warsaw University of Technology
Warsaw, Poland
and
the Faculty of Applied Information Technology
the University of Information Technology and Management
Rzeszów, Poland *homenda@mini.pw.edu.pl*

Agnieszka Jastrzębska
the Faculty of Mathematics and Information Science
the Warsaw University of Technology
Warsaw, Poland *A.Jastrzebska@mini.pw.edu.pl*

Mariusz Wrzesień
the Faculty of Applied Information Technology
the University of Information Technology and Management
Rzeszów, Poland *mwrzesien@wsiz.edu.pl*

Abstract

This work is a contribution to the field of time series classification. We propose a novel method that transforms time series into multi-channel images, which are then classified using Convolutional Neural Networks as an at-hand classifier. We present different variants of the proposed method. Time series with different characteristics are studied in this paper: univariate, multivariate, and varying lengths. Several selected methods of time-series-to-image transformation are considered, taking into account the original series values, value changes (first differentials), and changes in value changes (second differentials). In the paper, we present an empirical study demonstrating the quality of time series classification using the proposed approach.

Keywords: Time Series, Classification, Convolutional Neural Networks

1. Introduction

This paper deals with a relatively new field of pattern recognition: time series classification. This task has been the subject of intense studies for over a decade now. However, due to the challenging nature of time series data, this time series classification still poses challenges both from theoretical and practical perspectives. Time series classification methods are necessary in multiple real-world domains such as human activity recognition, computer and sensor-aided medical diagnosis systems, incident detection, and more. The essential aspect of time series classification is knowledge representation, which in this context can be translated to temporal data representation.

In this work, we present a novel contribution to the field of knowledge representation and time series classification. We elaborate on a method that transforms time series into two-dimensional images and then classifies obtained images. We use a Convolutional Neural Network (CNN) as an at-hand classifier. The main novelty and the key contribution presented in this paper are the methods for time-series-to-image transformation. Thus, the paper fits into the scope of the ISD conference (track T3) by elaborating on both knowledge representation methods and machine intelligence.

In the empirical section of this paper, we study time series with different characteristics: univariate, multivariate, and varying-length processed with our approach. We present several selected methods of time-series-to-image transformation that achieved the most promising results. These methods take into account the original series values, value changes (first differentials), and changes in value changes (second differentials). We should emphasize that the approach's transformation issue was not exhausted in this paper. The area allows for new developments and further research.

2. Literature Review

There are many algorithms for time series classification. We can group these methods into two main categories: those based on distance measures and those based on features describing time series. Sometimes, a third category is distinguished based on theoretical models of time series. However, the algorithms in this group can usually be classified into one of the two previously mentioned groups. The approach proposed in this paper belongs to the group of methods based on time series features. The features for describing time series are extracted from images that are an intermediate form of data representation in our case.

In the case of methods based on feature extraction, the classification step is done based on data present in the form of feature vectors. Such data representation can be processed using a conventional classifier such as a neural network or a decision tree. This is an extensive group of approaches since there is a wide range of different time series feature extraction methods. We can find feature extraction domains rooted in the frequency domain in this group. For example, methods based on spectral analysis, such as discrete Fourier transform [4], discrete wavelet transform [8] or shapelet transforms [3], can be mentioned here.

The group of feature-based methods is also rich with approaches that turn the data into sequences of symbols, then partition series into words and proceed with time series as with text data that needs to be classified. In this family of approaches, we can find methods named BOSS [11], Contractable BOSS [9], and WEASEL [12].

From the point of view of the study addressed in this paper, the most relevant approaches are the ones relying on a graphical representation of data. In this line of research, the classification procedure is envisioned as a pipeline made of three steps: conversion of time series into images, extraction of features from images, classification. Researchers working on such approaches focus on delivering new methods for time series imaging. For example, Hatami et al. [6] used recurrence plots. Zhang et al. [14] used a variant of recurrence plots named multi-scale Signed recurrence plot. After the transformation into images, feature extraction and image classification takes place. In recent papers, authors tend to merge these two steps together and apply a neural network to perform these tasks. Qin et al. [10] adopted a fusion residual network. A popular technique is to apply a Convolutional Neural Network, as did, for instance Garcia-Ceja et al. [5] or Sun et al. [13].

3. Time-Series-to-Image Transformations – Description of the Proposed Solution

Our design assumption was that the prepared solution should enable classification of time series regardless of their nature. That is, it should allow for classification of both univariate and multivariate series of varying lengths. In the described solution, the topic of missing values was not pursued, although various approaches were studied, for instance the method of Honaker and King [7] and we processed only the data with complete information.

In the final stage, we classified time series using CNN. Architecture of our model is presented in the sequential form (Figure 1). It uses two sets of convolutional and max pooling layers followed by few fully connected layers with ReLU as the activation function.

The last layer of the model was set to use softmax activation function. In order to prevent

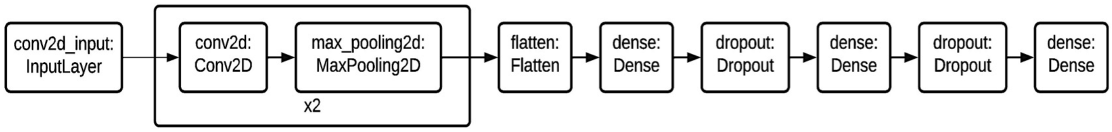

Fig. 1. Architecture of presented Convolutional neural network.

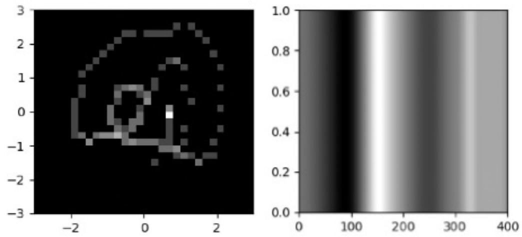

Fig. 2. Sample images obtained from transformation of time series: Values vs. Value Changes (left column) and Replicated Series of Values (right column). The examples illustrate the nature of the images obtained.

our network from overfitting, we used two dropout layers with probability coefficient equal to 0.15. The essential solution delivered in this paper concerns the prior transformation of the time series into a two-dimensional image of the size $w \times h$ ($width \times height$). For multivariate time series, each dimension adds one more "channel" to the image depth (denoted as d).

During our research, we have created and evaluated multiple time-series-to-image transformations. In this paper, we present only two best performing transformations. The first transformation is presented in Section 3.1. The second transformation is outlined in Section 3.2. Sample images obtained from the transformations are displayed in Figure 2. The examples illustrate the nature of the images obtained. The images were created as transformation of chosen elements of the Character Trajectories dataset.

3.1. Values, Value Changes

The input parameters for all the presented variants are the width w and the height h of the generated images and the color scale count c, which will be responsible for the color contrast in the image. In addition, the parameter d, the number of images, is determined automatically and depends on the number of dimensions of a given series.

In the case of the first variant, based on primary data, i.e., based on source time series stored in an array of length n, the following preprocessing is performed for each dimension (out of d) of the given time series creating:

- time series values: $a_1, a_2, a_3, a_4 \ldots, a_n$,
  value changes: $da_2 = a_2 - a_1, da_3 = a_3 - a_2, da_4 = a_4 - a_3 \ldots, da_n = a_n - a_{n-1}$,
- the following sequences of data pairs are considered: (a_k, da_k) where $k = 2, 3, 4, \ldots, n$
- the preprocessed values are converted into d images (two dimensional arrays of pixels) for each sequence of pairs, with assumed width w and height h, i.e., of size $w \times h$. The values and value changes are re-scaled to the integer values from the intervals $[0, w]$ and $[0, h]$. Let denote by i_k and j_k rounded values of a_k and da_k, respectively. Each entry (i_k, j_k) of the arrays, $k = 2, 3, \ldots, n$, is incremented by 1 (unless it gets the valued c).

Let us call this transformations Val/ValChng.

Notice that when the value (grayscale count) of the given pixel of the generated image reaches the defined maximum value c, then the value of this pixel is not increased any more.

This restriction comes from observations made in preliminary studies that a large portion of the pairs of values of the first differential are close to the point $[0, 0]$. This effect significantly increases the grayscale (or color) values, negatively affecting the ability to differentiate images.

Finally, the array of dimensions $w \times h \times d$ filled with integer values from the interval $[0..c]$ is then passed as input to a neural network-based classifier.

For this type of transformation, the generated images have similar natures. However, after a closer analysis, differences between the classes are visible. This observation suggests that neural network learning to distinguish classes based on differences should “see” these differences.

3.2. Replicated Series of Values

Based on the preliminary experiments done, we decided to prepare images that should be better suited to the learning algorithm used, i.e., to the convolutional networks. The approach also brings advantages due to the great possibilities in applying filters. For example, the differentiation operation would correspond to a convolution with a filter $[-1, 1]$.

The image generation in this variant is based on the source data without performing any further transformations: $a_1, a_2, a_3, \ldots, a_n$. The image prepared will be of size $n \times 1$, where n is the length of the series, i.e., it is a vector of values. Then this vector is replicated in order to create an image of size $n \times n$. Thus, the images will consist of vertical bars whose color/shade of gray will correspond to the (scaled) value of the series at a given time point. Let us call the above variant: ReplVal.

Notice that height of the initial image is 1. Hence vertical bars of the images are generated in a square plot by replicating the initial image that produces vertical stripes, cf. right column of Figure 2. Analyzing the images, one can see differences in the position, the width of the vertical stripes, and the gradients in the transitions between the light and dark areas. Therefore, we can assume that the classifiers could learn the differences between the classes on this basis.

4. Experimental Analysis

In order to test the proposed approach, transformations described in the previous section and the presented CNNs were implemented in the Python programming language. The experiments were executed on benchmark data taken from the repository [1] and are listed in Table 1.

4.1. Experiments

In line with the proposed approach in the previous paragraphs, tests were carried out for the mentioned time series. We decided to conduct experiments on relatively shallow models in which the layers related to feature extraction and classification are clearly visible.

The experiments were carried out as follows. For each dataset, transformations were carried out with each one of the presented time-series-to-image conversion models and neural networks were trained on train sets. Then, trained models were applied to test sets.

4.2. Results of Experiments

Table 1 shows classification accuracy achieved for ten datasets. While reviewing the values in the table, it can be noticed that the accuracy for the first variant was the best in six cases (out of 10). However, the average accuracy is the highest for the ReplVal transformation model, cf. Table 1. When we look into the nature of the datasets in retrospection to the achieved results, the proposed solutions turned out to be most successful for multidimensional datasets. Furthermore, longer time series datasets are typically classified more accurately.

The results of other researchers working on different image-based time series classification methods do not include information concerning all the presented benchmark datasets. For this

Table 1. Comparison of results obtained using two different image-based methods for time series classification on 10 selected data sets (accuracy in %). Obtained results are confronted with those published on the website [1] for 14 different classifiers. For each dataset taken is the worst and the best results among all 14 classifiers and average of all 14 classifiers. Only comparisons between the two image-based methods are indicated in bold.

	Val ValChng	ReplVal	Other classifiers min	Other classifiers max	Other classifiers ave
Basic Motions	**100.00**	92.50	not available		
Cricket	**97.22**	90.28	59.05	86.09	77.56
Crop	57.39	**76.48**	65.31	79.33	73.40
Earthquakes	64.75	**74.82**	71.70	74.96	74.34
Electric Devices	65.43	**67.23**	79.31	89.32	85.13
Freezer Regular Train	**99.79**	94.11	94.24	99.93	98.73
House Twenty	**96.64**	72.27	81.06	97.87	93.45
Japanese Vowels	52.70	**96.49**	not available		
Plane	**99.05**	98.10	98.83	100.00	99.77
Refrigeration Devices	**55.47**	48.27	61.16	79.09	73.42
Average	78.84	**81.05**	76.33	88.32	84.47

reason, we did not include in this paper such a comparison.

5. Conclusions

The presented method for time series classification was based on an intermediate step of time-series-to-image transformation. Convolutional Neural Networks were used for obtained images. The following time series representations were considered: original series values and value changes (first differentials). The following pairs of variables were considered: (i) values and values changes and (ii) (a row of) values replicated in order to create a square image. Based on these pairs images were generated.

The quality of the outcomes varies greatly and depends on a dataset, from almost 100% effectiveness for Basic Motions dataset to a weak 55% for Refrigeration Devices. Results published in [2] show that outcomes obtained in our initial experiments are comparable, besides those for Refrigeration Devices dataset. The best transformation out of two considered cannot be indicated, although in four out of ten randomly selected data sets, the accuracy measure for Val/ValChng transformation reaches almost 100%. It is worth emphasizing that there are no universal pattern recognition and classification methods. The application of some method on one dataset may not be very effective on another one.

Keeping in mind that our research on data transformations is at an initial stage, we plan to continue studies on various time series representation methods. A more extensive list of transformations needs to be considered. Based on the observation that the transformation using only original values (ReplVal) was the best one on average, cf. the last line of Table 1, we believe that a more detailed study is necessary into approaches utilizing original values.

Acknowledgment

This research was supported by the National Science Centre, grant No. 2019/35/D/HS4/01594, decision no. DEC-2019/35/D/HS4/01594.

The authors would like to thank Mr. Wiktor Rzemek, Mr. Jakub Stefaniuk and Mr. Przemysław Wcisło. They conducted computational tests in October-December 2021 in frames of a student's project under the supervision of Prof. Władysław Homenda.

References

1. http://timeseriesclassification.com/index.php
2. http://timeseriesclassification.com/results.php
3. Arul, M., Kareem, A.: Applications of shapelet transform to time series classification of earthquake, wind and wave data. Engineering Structures **228**, 111564 (2021). https://doi.org/https://doi.org/10.1016/j.engstruct.2020.111564
4. Bai, B., Li, G., Wang, S., Wu, Z., Yan, W.: Time series classification based on multifeature dictionary representation and ensemble learning. Expert Systems with Applications **169**, 114162 (2021). https://doi.org/https://doi.org/10.1016/j.eswa.2020.114162
5. Garcia-Ceja, E., Uddin, M.Z., Torresen, J.: Classification of recurrence plots' distance matrices with a convolutional neural network for activity recognition. Procedia Computer Science **130**, 157–163 (2018). https://doi.org/https://doi.org/10.1016/j.procs.2018.04.025, the 9th International Conference on Ambient Systems, Networks and Technologies (ANT 2018) / The 8th International Conference on Sustainable Energy Information Technology (SEIT-2018) / Affiliated Workshops
6. Hatami, N., Gavet, Y., Debayle, J.: Bag of recurrence patterns representation for time-series classification. Pattern Analysis and Applications **22**(3), 877–887 (Aug 2019). https://doi.org/https://doi.org/10.1007/s10044-018-0703-6
7. Honaker, J., King, G.: What to do about missing values in time series cross-section data. American Journal of Political Science **54**, 561–581 (2010). https://doi.org/https://doi.org/10.1016/j.procs.2015.05.444
8. Li, D., Bissyande, T.F., Klein, J., Traon, Y.L.: Time series classification with discrete wavelet transformed data. International Journal of Software Engineering and Knowledge Engineering **26**(09n10), 1361–1377 (2016). https://doi.org/https://doi.org/10.1142/S0218194016400088
9. Middlehurst, M., Vickers, W., Bagnall, A.: Scalable dictionary classifiers for time series classification. In: Intelligent Data Engineering and Automated Learning – IDEAL 2019, pp. 11–19. Springer International Publishing (2019). https://doi.org/https://doi.org/10.1007/978-3-030-33607-3_2
10. Qin, Z., Zhang, Y., Meng, S., Qin, Z., Choo, K.K.R.: Imaging and fusing time series for wearable sensor-based human activity recognition. Information Fusion **53**, 80–87 (2020). https://doi.org/https://doi.org/10.1016/j.inffus.2019.06.014
11. Schäfer, P.: The boss is concerned with time series classification in the presence of noise. Data Mining and Knowledge Discovery **29**(6), 1505–1530 (Nov 2015). https://doi.org/https://doi.org/10.1007/s10618-014-0377-7
12. Schäfer, P., Leser, U.: Fast and accurate time series classification with weasel. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. pp. 637–646. CIKM '17, ACM, New York, NY, USA (2017). https://doi.org/https://doi.org/10.1145/3132847.3132980
13. Sun, J., Takeuchi, S., Yamasaki, I.: Prototypical inception network with cross branch attention for time series classification. In: 2021 International Joint Conference on Neural Networks (IJCNN). pp. 1–7 (2021). https://doi.org/https://doi.org/10.1109/IJCNN52387.2021.9533440
14. Zhang, Y., Hou, Y., Zhou, S., Ouyang, K.: Encoding time series as multi-scale signed recurrence plots for classification using fully convolutional networks. Sensors **20**(14) (2020). https://doi.org/10.3390/s20143818