TIME SERIES CLASSIFICATION BASED ON VISUALIZATION OF RECURRENCE PLOTS

Lyudmyla Kirichenko, Petro Zinchenko

Abstract. Ordered data sets such as time series are found in almost all areas of human activity from cardiograms and to cyberattacks. Classification of time series is one of the most difficult tasks in data mining. In the article, a new method of time series classification based on the construction of recurrence plots is considered. The time series is transformed into a matrix, which characterizes the recurrence of the time series states, and the matrix is presented as a black-and-white image. Further, the convolutional neural network is used to classify the image. The application of the method is demonstrated by examples of simulated time series. A comparative analysis of the classification of noisy time series is carried out. The dependences of the classification accuracy on the noise level of time series are obtained. The results showed that the considered method has a high enough classification accuracy at high noise levels.

Keywords: Time series, noise, classification, recurrence plot, convolutional neural network

1 Introduction

Most of the processes occurring in the human body, nature, society, science and technology are complex, partly or completely random and have non-linear relationship. In practice, processes are presented in the form of corresponding time series, the properties of which make it possible to judge the properties of the generating process. The task of time series classification is one of the most difficult tasks of data mining. There are a number of approaches to the classification of time series, most of which are based on the calculation of various metrics between time series [1-5].

In the last few years, a number of studies have appeared in which the method of recurrent plots is used to classify time series. The recurrence analysis is based on such a property of the process as state repeatability, i.e. recurrence. In this case the recurrent properties of a time series are represented in the form of geometric structures and allow you to visualize the dynamics of the series. Methods of recurrence analysis were originally proposed in [6].

Over the past years, the recurrence plot method has been widely used for analyzing stochastic time series of various nature [7-11]. With the development of machine learning, recurrence characteristics calculated from time series began to be used as features for classification tasks [12-14].

Another approach to the application of recurrence methods for classification is the time series recognition directly from the images of recurrence plots. Since the best tools for recognition and classification of images are deep neural networks, a number of researchers use them to classify recurrence plots [4,15].

However, since such studies are fairly new, there has still not been enough attention paid to to classify noisy time series. The purpose of the presented work is to conduct a comparative classification of noisy time series based on the visualization of recurrence plots.

2 Method of recurrence plots

In recent decades, the traditional methods for studying time series has been significantly replenished with the methods of the theory of nonlinear dynamics and chaos. In this case, the time series is considered as the evolution trajectory of some nonlinear system. The main point of application of nonlinear dynamics methods to the analysis of the dynamic system trajectory is what the system attractor, containing all the information about the dynamics and properties of the system, can be restored by only time realization [16,17].

Recurrence analysis is one of such nonlinear dynamics methods used for time series and it is a tool for detecting not obvious dependencies in the dissipative dynamics. A recurrence analysis investigates the m-dimensional trajectory of a pseudo-phase space constructed by time realization. The well-known Packard-

Tackens procedure [16] for constructing a pseudo-phase space from only realization allows one to restore the attractor of a dynamic system:

$$F(t) = [x(t), x(t+\tau), ..., x(t+m\tau)],$$

where $F(t)$ is m-dimensional pseudo-phase space, $x(t)$ is time realization, τ is delay time.

In turn, the recurrence plot is a projection of m-dimensional pseudo-phase space onto a plane. Let the point x_i corresponds to the phase trajectory $x(t)$ that describes the dynamical system in m-dimensional space at the time moment $t = i$ for $i = 1,...,N$ then the recurrence plot *RP* is an array of points where a nonzero element with coordinates (i, j) corresponds to the case when the distance between x_j and x_i less ε:

$$RP_{i,j} = \Theta(\varepsilon - \| x_i - x_j \|), \quad x_i, x_J \in R^m, \, i, j = 1,...N,$$

where ε is the neighborhood size of the point x_i, $\left\| x_i - x_j \right\|$ is distance between points, $\Theta(\cdot)$ is Heaviside function.

An important step in the construction of the recurrence plot is the choice of the distance metric. The most popular is Euclidean metric, where the shape of the neighborhood is a circle of radius ε and the maximum norm, where the shape of the neighborhood is a square with a side ε. In many cases, the choice of norm is not fundamental, but for each specific task it makes sense to experiment. The obvious fact is that for homogeneous series, the Euclidean norm will be suitable, and in the case of heterogeneous, sharply changing series, the maximum norm for which the neighborhood has a large area is more appropriate.

In this work, when constructing recurrence plots, we used a one-dimensional phase space $m = 1$, which allows us to significantly reduce the constructing time and the Euclidean metric.

3 Convolutional neural network

Convolutional neural network (CNN) is a special architecture of artificial neural networks, aimed at efficient pattern recognition [18,19]. It is a prototype of the visual cortex. The visual cortex has the so-called simple cells that respond to straight lines at different angles, and complex cells, the reaction of which is associated with the activation of a specific set of simple cells. For example, some neurons are activated when they perceive vertical border, and some are horizontal or diagonal. All these neurons together form a visual perception. The idea that specialized components solve specific problems (like cells of the visual cortex that look for specific characteristics) is used in machine learning.

Thus, the idea of convolutional neural networks is to alternate convolutional layers and sub-sampling layers. The network structure is unidirectional (without feedbacks), fundamentally multilayer. For training, standard methods are used, most often the back propagation method of error. The function of activation of neurons can be different, according to the task.The architecture of the network got its name because of the convolution operation, the essence of which is that each image fragment is multiplied by the matrix (core) of the convolution element by element, and the result is summed and written to the same position in the output image.

The network works as follows. An image passes through a series of convolutional, nonlinear layers, union layers, and fully connected layers, and output is generated. The conclusion may be the class or probability of the classes that best describe the image.

The first layer in the CNN is always convolutional. It is a set of feature cards (these are ordinary matrices), each card has a synaptic core (scanning core or filter). The size of all cards of a particular convolutional layer is the same.

The core is a filter or window that slides over the entire area of the previous map and finds certain signs of objects. For example, if the network was trained on faces, then one of the cores during the learning process would give the greatest signal in the area of the eye, mouth, eyebrow or nose, the other core could reveal other signs. The size of the core is usually taken in the range from 3x3 to 7x7. If the size of the nucleus is small, then it will not be able to highlight any signs; if it is too large, the number of connections between neurons increases. Also, the kernel size is chosen so that the size of the convolutional layer cards is even, this allows you to not lose information when reducing the dimension in the subsample layer, described below.

When a picture passes through one convolutional layer, the output of the first layer becomes the input value of the 2nd layer. After applying a set of filters after the first layer, filters that represent higher level properties will be activated. The types of these properties can be half rings (a combination of a straight border with a bend) or squares (a combination of several straight edges). The more convolutional layers an image goes through and the further it moves across the network, the more complex the characteristics are displayed in the feature maps.

After convolutional layers, a pooling layer follows. It is also referred to as a downsampling layer. In this category, there are also several layer options, with maxpooling being the most popular. This basically takes a filter (normally of size 2x2) and a stride of the same length. It then applies it to the input volume and outputs the maximum number in every subregion that the filter convolves around.The last type of layer is the layer of an ordinary multilayer perceptron. The purpose of the layer is classification, it models a complex nonlinear function, optimizing which improves the quality of recognition.The output layer is connected to all neurons of the previous layer. The number of neurons corresponds to the number of recognized classes.

Now we are ready to describe the overall architecture of our CNN. As depicted in Fig.1, the net contains eight layers with weights; the first five are convolutional and the remaining three are fully-connected.

The output of the last fully-connected layer is fed to a 2-way softmax which produces a distribution over the 2 class labels.The neurons in the fully-connected layers are connected to all neurons in the previous layer. Max-pooling layers, follow second and fourth convolutional layer. The ReLU non-linearity is applied to the output of every convolutional and fully-connected layer.

The first convolutional layer filters the 256×256×1 input image with 8 kernels of size 5×5. The second convolutional layer takes as input the output of the first convolutional layer and filters it with 16 kernels of size 4 × 4. The third convolutional layer has 32 kernels of size 3 × 3 connected to the outputs of the second convolutional layer. The fourth convolutional layer has 64 kernels of size 3 × 3. The fully-connected layers have 1024 neurons each. For training the network was used Adam is an adaptive learning rate optimization algorithm.

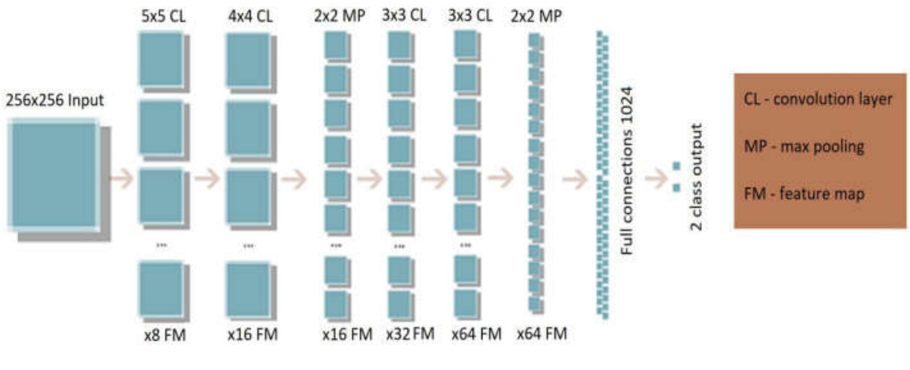

Fig. 1. Architecture of our convolutional neural network

4 Description of the experiment and results

As input time series in the work, we selected sinusoid time realizations with different periods of oscillation and different degrees of noise. Such series are typical models of real processes. We can present the time realization as the sum of a sinusoid component and a noise one: $X(\mathrm{t}) = Y(\mathrm{t}) + \mathrm{z}(\mathrm{t})$,where Y(t) is time series, z(t) – additive noise. As a value characterizing the ratio of signal to noise, the coefficient Snr was used $Snr = S[\mathrm{Y(t)}]/S[\mathrm{z(t)}]$, where S is standard deviation. By changing the coefficient Snr we specify a different degree of noise in the time series.

To carry out the classification, the input time series were split into two classes. The first class consisted of sinusoids, for which the frequencies varied in the range $f2 \pm fR$, for the second class the frequency range was $f2 \pm fR$. The frequency choice to the sine wave from the ranges $f1 \pm fR$ and $f2 \pm fR$ was carried out randomly. The values $f1$, $f2$, fR , and $Fdist = |f1 - f2|$ varied during the experiment.

Fig. 2 shows plots of noisy sinusoids with different frequencies and different noise degree. In this case, the length of the time series is 256 values. At the top of the Fig. 2, sinusoids from the lower frequencies class with parameter Snr values = 1, 0.7 and 0.4 are presented. The bottom of Fig. 2 shows examples of sine waves from the second class with the same values Snr .

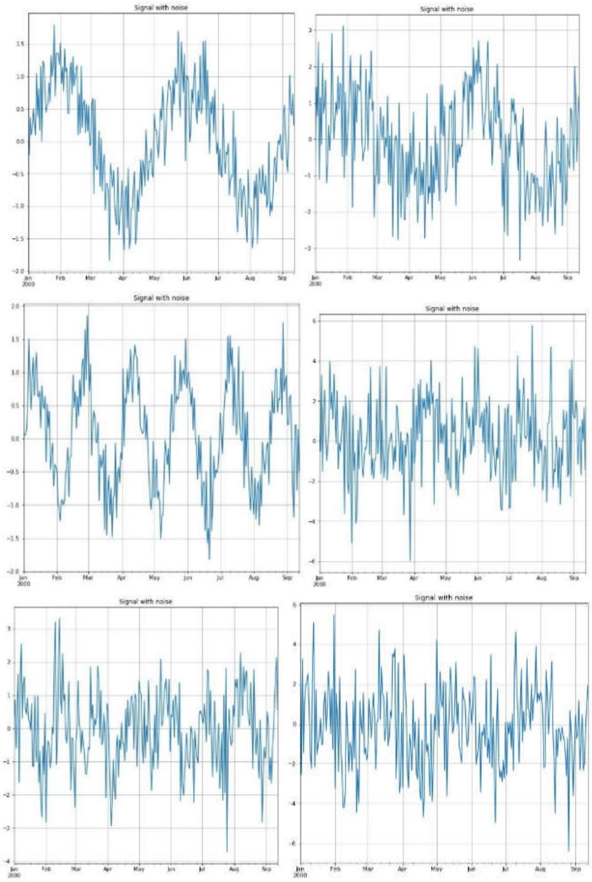

Fig. 2. Noisy sinusoids of both classes with Snr = 1, 0.7 and 0.4

Fig. 3 shows the recurrence plots corresponding to the time series of both classes. On the left recurrence plots for time series without noise are shown, and on the right ones with noise at Snr = 1 are presented. When classifying, the training sample consisted of 200 time series of two classes (100 for one class and 100 for another), each of length 256 values. The test sample also included 200 time series. Such values were chosen in order to get the experimental conditions closer to typical real datasets.

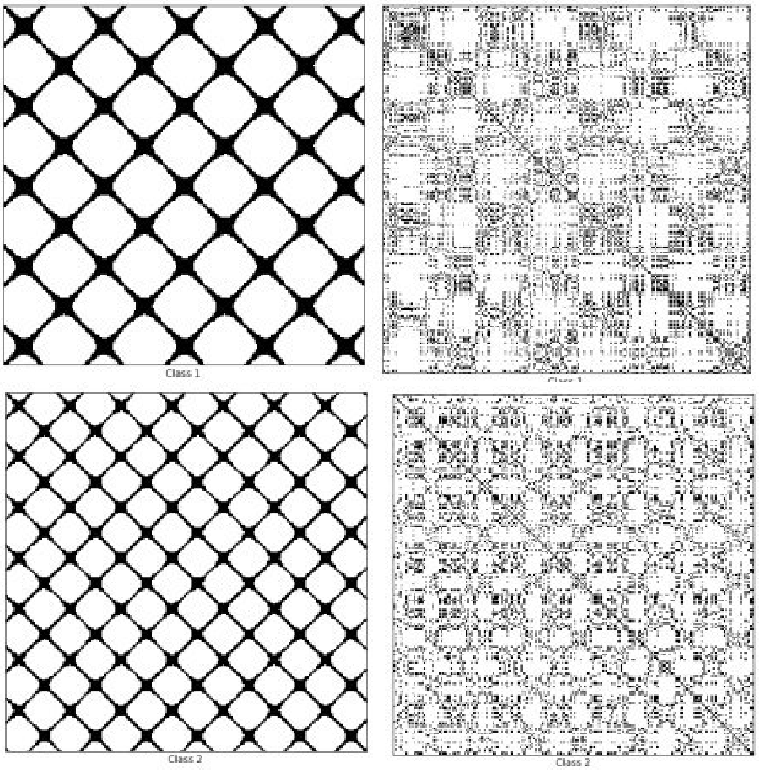

Fig. 3. Recurrence plots of time series without noise and with noise

Previously, recurrence plots of each time serie were obtained for input to the neural network. Thus, we have moved from time series to images that a neural network should recognize.A numerical experiment was conducted for different values of the parameters $f1$, $f2$, fR , $Fdist$ and Snr . Reseach have shown that without noise, the classes are distinguishable with an accuracy of 100% even when frequency ranges had a common boundary at the value $Fdist = 0$.

The main attention during the experiment was paid to increasing the noise level of the time series, i.e. reduction ratio Snr . The classification results showed very good accuracy at noise level $Snr > 0.6$. It should be noted that when decreasing Snr , the training time of the neural network (the number of epochs) increased from about 10 to 30. It is easily explained by the complexity of recognition at low values Snr .

Fig. 4 presents a part of the test of recurrence plot sample that were fed to the input of the classifier with noise level Snr =0.7. This sample is already classified by the neural network. At the top there are recurrence plots from the class with lower frequencies, and at the bottom there are ones from the second class.

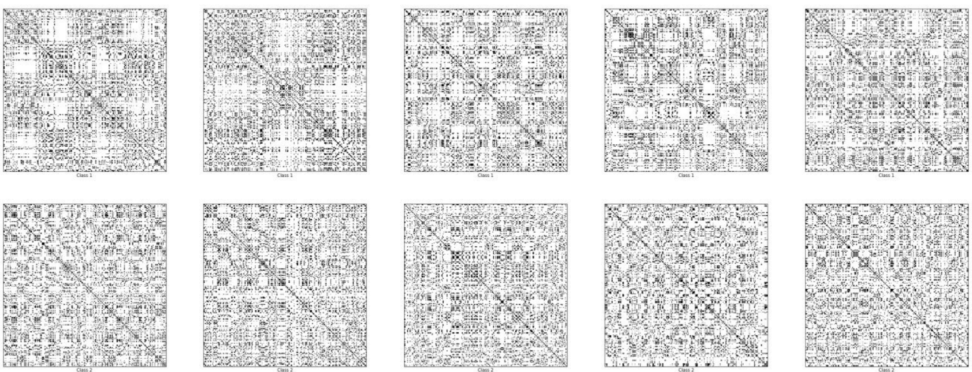

Fig. 4. Classified recurrence plots of time series of both classes at Snr =0.7

Table 1 presents the classification accuracy and the number of epochs depending on the noise level Snr. It is worth noting that with an increase in size of the training sample to 400 values, the classification accuracy at ratio =0.4 Snr increases to 0.811.

Table 1. Classification accuracy and number of epochs

Noise level Snr	Accuracy	Numberofepochs
1	0.992	10
0.7	0.967	14
0.6	0.945	17
0.5	0.778	24
0.4	0.66	29

5 Conclusion

In the work, a method for classifying time series based on the construction of recurrence plots using the simple architecture of a convolutional neural network have been investigated. A comparative analysis of the classification of noisy time series was carried out. The dependences of the classification accuracy on the noise level were obtained. The results showed that the considered method has a fairly high classification accuracy even with a large degree of noise.The results of the work can be used for the classification of time series of stochastic type by machine learning methods. In our future research we intend to concentrate on classifying real time series from known datasets.

References

1. Esling P., Agon C.: Time series data mining. ACM Computing Surveys 46(1) (2012).
2. Ben D: Feature-based time-series analysis, https://arxiv.org/abs/1709.08055, last accessed 2019/08/20.
3. Bagnall A., Lines J., Bostrom A., Large J., Keogh E.: The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data Mining and Knowledge Discovery 31 (3), 606-660 (2017).
4. Hassan Ismail,FawazGermain,Forestier Jonathan, Weber Pierre-Alain Muller Deep learning for time series classification: a review Data Mining and Knowledge Discovery 2019, Volume 33, Issue 4, pp 917–963.
5. KrisztianBuzaTimeSeries Classification and its Applications https://doi.org/10.1145/3227609.3227690 last accessed 2019/08/20
6. J.P. Eckmann, S.O. Kamphorst, and D. Ruelle. "Recurrence Plots of Dynamical Systems." EPL (Europhysics Letters), vol. 4, no. 9, 1987, pp. 973-977.
7. N. Marwan, N. Wessel, U. Meyerfeldt, A. Schirdewan and J. Kurths. "Recurrence-plots-based measures of complexity and application to heart-rate-variability data", Physical Review E, vol. 66, no. 2, pp. 026702-1 - 026702-6, 2002.
8. N. Marwan, M. Romano, M. Thiel, and J. Kurths, "Recurrence plots for the analysis of complex system." *Physics Reports*, vol. 438, no. 5-6, pp. 237-329, 2007.
9. J.P. Zbilut, J.M. Zaldivar-Comenges and F. Strozzi. "Recurrence quantification based Liapunov exponent for monitoring divergence in experimental data." *Physics Letter* A, Vol. 297, issue 3-4, pp. 173-181, 2002.
10. E.J. Ngamga, A. Nandi, R. Ramaswamy, M.C. Romano, M. Thiel, and J. Kurths, "Recurrence analysis of strange nonchaotic dynamics." *Physical ReviewE*, vol. 75, issue 3, pp. 036222, 2007. doi:10.1103/PhysRevE.75.036222.

11. L.O. Kirichenko, Y. Kobitskaya, and A. Habacheva. "Comparative Analysis of the Complexity of Chaotic and Stochastic Time Series." Radioelectronics. Informatics. Management, no. 2(31), pp. 126-134, 2014.
12. L. Kirichenko, T. Radivilova and V. Bulakh, "Classification of Fractal Time Series Using Recurrence Plots," 2018 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), Kharkiv, Ukraine, 2018, pp. 719-724. doi: 10.1109/INFOCOMMST.2018.8632010
13. Thilo Michael, Stephan Spiegel, SahinAlbayrak Time Series Classification using Compressed Recurrence Plots http://www.dai-labor.de/fileadmin/Files/Publikationen/Buchdatei/Published.pdf last accessed 2019/08/20.
14. Nima Hatami, Yann Gavet, Johan DebayleBag of Recurrence Patterns Representation for Time-Series Classificationhttps://arxiv.org/abs/1803.11111v1last accessed 2019/08/20.
15. Nima Hatami, Yann Gavet, Johan DebayleClassification of Time-Series Images Using Deep Convolutional Neural Networks https://arxiv.org/abs/1710.00886 last accessed 2019/08/20
16. F. Takens, D.A. Rand, and L.-S. Young. "Detecting strange attractors in turbulence." Dynamical Systems and Turbulence: Lecture Notes in Mathematics,Springer-Verlag, vol. 898, pp. 366–381, 1981.
17. Iwanski J. S., Bredley E. Recurrence plots of experimental data: To embed or not to embed? Chaos, №8(4), 1998 P. 861-871
18. Y. LeCun and Y. Bengio: Convolutional Networks for Images, Speech, and Time-Series, in Arbib, M. A. (Eds), The Handbook of Brain Theory and Neural Networks, MIT Press, 1995
19. Ciresan, Dan; Ueli Meier; Jonathan Masci; Luca M. Gambardella; JurgenSchmidhuber "Flexible, High Performance Convolutional Neural Networks for Image Classification" Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence-Volume 2: 1237–1242. 2013 http://people.idsia.ch/~juergen/ijcai2011.pdf last accessed 2019/08/20