Classification of SAX-like Time Series Bitmaps with Siamese Convolutional Neural Networks

Mariusz Wrzesień
University of IT and Management, Faculty of Applied Informatics
Rzeszów, Poland *mwrzesien@wsiz.edu.pl*

Miłosz Wrzesień
AGH University of Science and Technology, Faculty of Electrical Engineering Automatics Computer Science and Biomedical Engineering
Kraków, Poland *milo.wrzesien@gmail.com*

Janusz Korniak
University of IT and Management, Faculty of Applied Informatics
Rzeszów, Poland *jkorniak@wsiz.edu.pl*

Abstract

This research paper concerns the classification of multiclass univariate time series. It extends the existing time series imaging method and describes the process of obtaining new transformations of this sort. The presented techniques leverage the well-known Symbolic Aggregate Approximation representation, which transforms time series from numerical to symbolic domain. The obtained symbolic approximations are later turned into images. After raw time series data is transformed into two-dimensional grayscale bitmaps, these bitmaps are used as input for two alternative deep learning classification approaches. Our study focuses on comparing regular Convolutional Neural Networks and Siamese Neural Networks as time series classifiers. Experimental studies and comparative analyses were performed on well-known, publicly available datasets.

Keywords: time series classification, time series imaging, siamese neural networks, convolutional neural networks.

1. Introduction

Analyzing of time series data has proven useful in many different scenarios. The constant interest in this field, as well as the rapid development of new time series solutions, motivated us to improve the existing Symbolic Aggregate Approximation (*SAX*) method with a Convolutional Neural Network (*CNN*) approach for time series classification [7],[1]. This paper also address the challenge of achieving reliable classification performance when the number of training samples is limited. In real-world applications, having a model that quickly produces reliable classification results based on a limited number of data samples is more important than achieving near-perfect results, which require extensive training time with large datasets. In certain instances, expanding the training dataset is nearly impossible, and data augmentation does not always improve model performance. To address these concerns, this paper contributes to the field of time series classification by:

- attempts to improve the classification performance and accuracy of *SAX CNN* method by incorporating Siamese Neural Networks as an alternative classifier and expanding the core time-series-to-image transformation,
- a comparative study of two neural network-based classifiers in time series classification, and low training sample sizes and training evaluation times.

One of the concepts of time series classification is the process of transforming raw time series data into an image. Homenda et al. [3] provide an example of such a study. The authors provide a similar set of time series classification approaches to the one used in this paper.

2. The method

The standard SAX CNN [7] is based on a two-step transformation of the time series. First, raw numerical values are being converted into symbolic and then visual representations. Then, a time series sample is transformed into a word of arbitrary length using the method described in the referenced paper [7]). The second step involves representing the resulting words as a grayscale visual grid. The generated graphical representation is classified using a shallow CNN.

This research focuses on generating a bitmap representation of an input time series. This paper proposes and verifies two new time series imaging approaches through experimental analysis and compares them to the (**standard SAX CNN**).

Sort SAX CNN. In this new approach, the letters of the output SAX word are sorted alphabetically before being transformed into an image.

Shift SAX CNN. In this new approach, the SAX word fills out vertically columns of the grid, while items in each subsequent column are shifted one position from the previous one in a circular fashion. Circular shifting means that the last item moves to the first place. The grid is then transformed into an image.

The prepared images are ready for training with a convolutional neural network. For this purpose the simple model shown in the figure 1 has been chosen. This model is a universal architecture for most datasets and proposed imaging methods. Moreover, the model avoids problems with overfitting that ResNet [6], [2] and AlexNet [5] face.

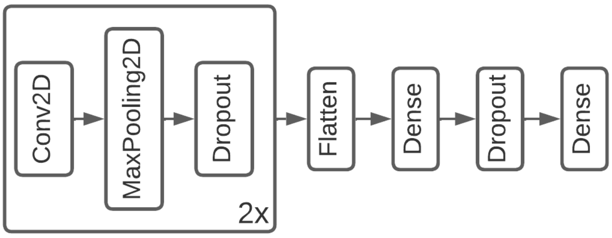

Fig. 1. The architecture of our CNN.

In addition to using a single standard CNN, we evaluated an alternative classification approach that uses Siamese neural networks (SCNNs) [4]. This architecture uses two twin convolutional neural networks. Both networks share an architecture that matches the one presented above (Figure 1).

Before performing the transform, all three time series imaging approaches require the selection of two hyperparameters. These parameters are word length, alphabet size, and square size. The first parameter defines the number of letters that form the SAX word, which is assumed to be a power of two for the *Standard* and *Sort* transforms, and any for *Shift* transform. Alphabet size is the number of different letters used to form the SAX word. It also directly impacts the number of different grayscale shades in an output image. The higher this number, the greater the range of colors in the image. The third parameter, *square size*, is insignificant, therefore it remains static in the experiments performed [7].

There are also hyperparameters specific to each classifier that need to be initialized. We ensured that the singular and twin CNNs used in the SCNN classifier shared the same hyperparameters, including the number of training epochs, the optimizer, the early stopping settings, and the batch size of the training data. The SCNN method uses an additional hyperparameter: the *number of nearest neighbors* used in the KNN-based prediction process.

To evaluate the performance and accuracy of the presented methods, we selected eight sample collections of time series from the website
`https://www.timeseriesclassification.com`.

Table 1. Comparison of CNN and SCNN for different numbers of training samples

train size percentage	30%		50%		75%		85%		Max accuracy	
dataset/ classifier	CNN	SCNN	CNN	SCNN	CNN	SCNN	CNN	SCNN	CNN	SCNN
BeetleFly	45,00%	85,00%	95,00%	40,00%	80,00%	55,00%	80,00%	45,00%	**95,00%**	85,00%
Coffee	67,86%	82,14%	85,71%	21,43%	46,43%	92,86%	89,29%	28,57%	89,29%	**92,86%**
Ham	82,86%	59,05%	79,05%	47,62%	81,90%	56,19%	78,10%	50,48%	**82,86%**	59,05%
Herring	59,38%	60,94%	59,38%	65,63%	59,38%	64,06%	59,38%	62,50%	59,38%	**65,63%**
HouseTwenty	60,50%	76,47%	70,59%	84,87%	70,59%	85,71%	68,07%	80,67%	70,59%	**85,71%**
Meat	66,67%	90,00%	33,33%	83,33%	33,33%	90,00%	56,67%	90,00%	66,67%	**90,00%**
Rock	60,00%	34,00%	54,00%	50,00%	84,00%	24,00%	70,00%	42,00%	**84,00%**	50,00%
UMD	61,81%	38,89%	73,61%	40,97%	78,47%	29,86%	81,94%	33,33%	**81,94%**	40,97%
Average Accuracy	63,01%	**65,81%**	**68,83%**	54,23%	**66,76%**	62,21%	**72,93%**	54,07%		

Table 2. Comparison of CNN and SCNN in terms of training time

	Standard		Sort		Shift	
dataset/ classifier	CNN	SCNN	CNN	SCNN	CNN	SCNN
BeetleFly	4,40	4,61	3,18	4,51	3,54	4,47
Coffee	4,95	4,48	3,34	4,21	3,51	4,87
Ham	9,19	7,10	5,15	7,90	10,66	9,49
Herring	3,29	4,78	3,20	4,90	3,57	5,88
HouseTwenty	14,03	5,20	6,63	6,88	4,40	7,82
Meat	3,99	4,58	4,61	5,27	5,29	12,60
Rock	5,07	4,01	3,30	5,42	4,98	7,82
UMD	6,17	5,01	7,38	6,40	8,08	17,01
Average time [s]	5,70	**4,93**	**4,32**	5,30	**5,57**	8,79

3. Performed experiments

Our comparison aimed to evaluate the performance and accuracy of the two classification approaches we implemented. To ensure reliable results, we repeated the training and prediction processes five times for each hyperparameter configuration. We also made sure that the computational resources provided for both methods were equal, as we ran the experiments on the same hardware.

The results show that *Standard SAX CNN* model achieved the best results with those parameters belonging to the following sets: $\{49, 64\}$ and $\{16, 18, 24\}$. The number of possible nearest neighbors needed to perform the prediction process by SCNN was arbitrarily set to $\{3, 5, 7\}$ at first, but after many evaluations on different datasets, we observed that looking at 3 nearest neighbors yielded us the best results most of the time.

Our experimental analysis included two additions that were not present in [7]. First, we tested the accuracy of our classifiers in an early classification setting, by providing the models with only a fraction of the available training samples. Second, we measured the exact training time of the classifiers to determine which model (CNN or SCNN) performs better in this setting.

Table 1 presents a comparison of the average accuracy using the *Standard SAX CNN* transformation by the two classifiers, CNN and SCNN. We summarised the average accuracy across all eight datasets to determine which classifier performs best with all of the chosen data samples. As mentioned above, we also measured how these models perform when we constrain the number of available training files. A clear advantage of the SCNN model over the regular CNN is observed when the number of training samples is constrained to 30%, as the SCNN demonstrates higher overall average accuracy (Table 1).

The second set of experiments focused on training time. Due to the limited number of training samples and the low resolution of the input images (ranging from $20x20$ to $108x108$), low training times were anticipated given the selected datasets. Consequently, even with a higher number of training epochs, the models learned in approximately 10 seconds. (Table 2). Interestingly, the SCNN classifier was able to complete the training process in a comparable or shorter time, while achieving superior accuracy on five out of eight data samples tested (see the 30% column in Table 1).

4. Conclusions

Based on the experimental results, no definitive winner emerges. However, it can be concluded that using a Siamese Convolutional Neural Network (SCNN) as a classifier for SAX-like time series transformations is advantageous when the number of training samples is limited and quick training is essential. When the full training set is available and time is not a constraint, the regular CNN classifier generally performs better. However, additional experiments are needed to test these methods with other datasets. Specifically, we need to clarify why in some cases the CNN method sometimes yields excellent results while the SCNN method yields poor ones for the same dataset, and vice versa for different datasets.

Because the chosen classifier is highly accurate, further extensions should focus on the time series imaging technique itself. Current transformations are limited to representing univariate time series samples. The next step should be to propose more flexible transformations and extending existing ones to higher dimensions. This would enable the classification of multivariate time series within a similarly structured pipeline.

References

[1] Dau, H.A., Bagnall, A., Kamgar, K., Yeh, C.C.M., Zhu, Y., Gharghabi, S., Ratanamahatana, C.A., Keogh, E.: The ucr time series archive. IEEE/CAA Journal of Automatica Sinica 6(6), pp. 1293–1305 (2019)

[2] Geng, Y., Luo, X.: Cost-sensitive convolutional neural networks for imbalanced time series classification. Intelligent Data Analysis 23(2), pp. 357–370 (2019)

[3] Homenda, W., Jastrzębska, A., Pedrycz, W., Wrzesień, M.: Time series classification with their image representation. Neurocomputing 573, pp. 127214 (2024)

[4] Koch, G., Zemel, R., Salakhutdinov, R.: Siamese neural networks for one-shot image recognition. https://www.cs.utoronto.ca/~rsalakhu/papers/oneshot1.pdf (2015)

[5] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Commun. ACM 60(6), pp. 84–90 (2017)

[6] Wang, Z., Yan, W., Oates, T.: Time series classification from scratch with deep neural networks: A strong baseline. In: 2017 International Joint Conference on Neural Networks (IJCNN). pp. 1578–1585 (2017)

[7] Wrzesien, M., Wrzesien, M., Homenda, W.: Time series classification using images. In: da Silva, A.R., da Silva, M.M., Estima, J., Barry, C., Lang, M., Linger, H., Schneider, C. (eds.) Information Systems Development, Organizational Aspects and Societal Trends (ISD2023 Proceedings) (2023)