Unsupervised Time Series Classification for Climate Data

Alex Romanova[1]

[1]Melenar, LLC, McLean, VA, US

Abstract

Outstanding success of Convolutional Neural Network image classification in the last few years influenced application of this technique to a variety of embeddable entities. CNN image classification methods are getting high accuracies but most of them are based on supervised machine learning that requires labeling of input data and do not interpret unknown data. In this study we introduce unsupervised machine learning model that maps embeddable entity pairs to symmetric or asymmetric images by transforming pairwise vectors to Gramian Angular Fields (GAF) images. Based on CNN transfer learning image classification model we indicate metrics of GAF images visual similarity to symmetric or asymmetric classes. Experimenting with climate data, we illustrate several scenarios that show that this model is reliable for entity pairs with one-way relationships but insufficient for entity pairs with two-way relationships.

1 Introduction

In the last few years deep learning demonstrated great success outperforming previous state-of-the-art machine learning techniques in various domains [12]. In particular, after the evolutionary model AlexNet was created in 2012, deep learning techniques became very powerful in Convolutional Neural Network (CNN) image classification [11].

Success stories of ImageNet deep learning influenced usage of CNN image classification as an instrument for other techniques such as vector classifications. One of such techniques was suggested by Ignacio Oguiza as a method of encoding time series to Gramian Angular Field (GAF) images and classify GAF images based on fast.ai library [8].

In his study author translated Olive Oil time series data to images and demonstrated that image classification works well for time series classification. Specifically, author used transfer learning technique that allowed to train model on small sets of training data. Fine-tuning a network with transfer learning works usually much faster and has higher accuracy than training image classification model from scratch [21].

Unsupervised classification method that we introduce in this paper follows both techniques mentioned above: encoding time series to GAF images and using transfer learning for image classification. What is completely different in our method is application of these techniques to unlabeled data. Our model determines whether two time series vectors are similar or not similar to each other. Method is build on generating pairs of similar and different vectors, and then transforming pairwise vectors to GAF images for classification.

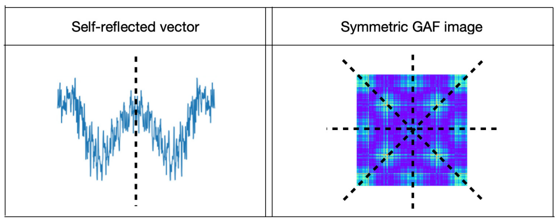

Figure 1: Self-reflected vectors transformed to GAF images create symmetric squares.

The illustration on Figure 1 shows that by transforming self-reflected, mirror vectors to GAF images we are creating symmetric squares. On the contrary, by transforming pairs of different vectors to GAF images we create asymmetric squares (see training data examples on Figure 2). The process of approximately classifying images to symmetric or asymmetric classes acts as a dimensionality reduction and therefore it is expected to work better

https://doi.org/10.7557/18.6250

than more complex image classification. To focus on symmetry we were inspired by geometric deep learning [4].

Unsupervised machine learning finds many types of unknown patterns in data and helps to find features which can be useful for categorization. It is easier to get unlabeled data from a computer than labeled data, which needs manual intervention. One of emerging areas of unsupervised machine learning is self-supervised learning where labeling is automated. In terms of terminology, it is still questionable what methods should be determined as unsupervised machine learning and what methods should be determined as self-supervised machine learning. As self-supervised machine learning technology mostly focuses on image or text classifications, we will call our novel techniques as an unsupervised machine learning.

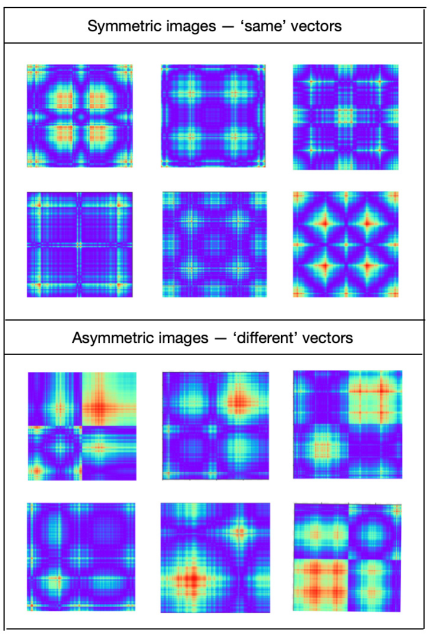
Symmetric images — 'same' vectors
Asymmetric images — 'different' vectors

Figure 2: Training data examples. Symmetric images demonstrate 'same' class for self-reflected vectors converted to GAF images. Asymmetric images demonstrate 'different' class for pairs of different vectors converted to GAF images.

The pairwise vector entity classification method that we will introduce in this paper has limitations. In particular, we will show that this method is insufficient for entity pairs with two-way relationships and therefore it should be used only for entity pairs with one-way relationships. Considering these limitations, this method is applicable to entities that can be transformed to vectors such as time series, sequences, and many other types of entities that are well described as the list of *2Vec entities in GitHub paper [14].

We used this technique to solve one of the Natural Language Processing tasks in our technical blog [17]. We classified word pairs by word similarities to identify word pairs that are unexpected to be next to each other in documents or data streams.

For experiments in this study we will use data about history of daily temperatures in cities and will compare temperatures between city pairs. Based on this data we will show how this model works for classification of entity pairs with one-way relationships and why this model is not reliable for classification of entity pairs with two-way relationships.

Our solution for time series classification:

- Convert daily temperature time series to vectors
- Create reflected pairwise vectors by concatenating pairs of vectors
- Transform pairwise vectors to GAF images
- Classify images using CNN transfer learning
- Examine how the model works for entity pairs with two-way and one-way relationships
- Prove the hypothesis that the model is not reliable for entity pairs with two-way relationships
- Demonstrate successful scenarios for entity pairs with one-way relationships.

2 Related Work

In the recent years deep learning techniques such as Long-Short Term Memory (LSTM) were successfully applied to time series forecasting applications [9]. CNN image classification techniques are very useful for time series classification especially when one-dimensional time series signals are transformed into two-dimensional matrices.

In study [20] authors suggest time series transformations to Gramian Angular Fields (GAF) and

Markov Transition Fields (MTF) images. Techniques of transforming vectors to two-dimensional matrices allow machines to “visually” recognize signals and improve time series classification. In one of our studies we showed higher accuracy metrics of CNN image classification model with vector to GAF image transformation than accuracy metrics of models with plot pictures [16].

Self-supervised representation learning has made significant progress over the last years, almost reaching the performance of supervised baselines on many downstream tasks [2]. The closest to our method are discriminative self-supervised methods or contrastive methods that are based on direct comparison between training samples [5].

As our method categorizes pairs of entities to classes of similar pairs and not similar pairs, it is conceptually comparable with a combination of Siamese Nets [3] and self-supervised contrastive methods [5]. Siamese nets were first introduced in 1993 by Bromley and LeCun to solve signature verification as an image matching problem [3].

Recently, CNN deep learning techniques has shown great promise for the analysis of atmospheric imaging and for the estimation of tropical cyclones and their precursors. In one of the first studies such CNN deep learning was applied to climate pattern recognition problems [1]. However the survey study [13] of deep learning and machine learning techniques applied to hydrological processes, climate change and earth systems, shows that deep learning for climate data mining is still in the first stages of development, and the research is still progressing.

3 Methods

For data processing, model training and interpreting the results we will use the following steps:

- Transform climate raw data to embedded vectors
- Create pairs of pairwise vectors: self-reflected, mirror vectors for 'same' class and concatenated different vectors for 'different' class
- Transform pairwise vectors to GAF images for CNN image classification
- Train CNN image classification model to distinguish symmetric and asymmetric images
- Classify entity pairs based on interpretation of trained model results.

Data preparation, training and interpretation techniques are described in details in our technical blog [19].

3.1 Transform Time Series Data to Pairwise Vectors

For unsupervised entity pair classification we will use the following steps:

- Concatenate entity vectors with themselves by reversing the second vector and label such vector pairs as 'same'
- Concatenate vectors of different entities by reversing the second vector and label such vector pairs as 'different'.

3.2 Transform Pairwise Vectors to GAF Images

As a method of vector to image translation we will use Gramian Angular Field (GAF), a polar coordinate transformation based technique [20].

3.3 Train CNN Image Classification Models

To deal with comparatively small set of training data, instead of training the model from scratch, we will follow ResNet-50 transfer learning: load the results of model trained on images from the ImageNet database and fine tune it with data of interest [8].

Fast.ai CNN transfer learning image classification can be used for both supervised and unsupervised machine learning. Python code for transforming vectors to GAF images and fine tuning ResNet-50 is described in fast.ai forum [7].

3.4 Use Results of Trained Models

To calculate how similar are vectors to each other we will combine them as pairwise vectors and transform to GAF images. Then we will run GAF images through trained image classification model. If probability of getting to the 'same' class is higher than 0.5, we will consider the vector pair as similar, otherwise as not similar. Also if 'same' probability is high (for example, more then 0.8) the pair can be considered as very similar.

4 Experiments

4.1 Transform Raw Data to Embedded Vectors

As raw data for this study we will use average daily temperatures in Celsius degrees for time period from January 1, 1980 to September 30, 2020 for 1000 most populous cities in the world taken from kaggle.com: "Temperature History of 1000 cities 1980 to 2020" [10].

To experiment with time series classifications, we will convert raw data to data set embedded vectors of the length 365 for average daily temperatures by city and by year and label vectors as 'city-year'.

The process of transforming raw data to vectors and python code are described in our technical blog [18]. Data preparation for pairwise vector method, model training and interpretation techniques are described in details in another post of our technical blog [19].

4.2 Transformation of Pairwise Vectors to GAF Images

For training data we will create 'same' class of symmetric images and 'different' class of asymmetric images. The Figures 1 and 2 illustrate that when converting self-reflected, mirror vectors to GAF images, we are getting symmetric images. Otherwise, when converting pairs of different vectors to GAF images, we are getting asymmetric images.

- For the 'same' class we will combine vectors with themselves reversing second vectors.
- For the 'different' class we will combine random pairs of vectors with temperatures of different years and different cities with reversed second vectors.

4.2.1 'Same' Class: Coalesce Vectors with Reversed Themselves

To generate training data images for the 'same' class we will combine vectors with reversed themselves. For each vector we will create a label 'city-year'. For vector pairs we will combine these labels to 'city-year-city-year' labels and will use these labels as file names. Please see Figure 3 as an example of 'same' class images. This figure illustrates that self-reflected mirror vector and corresponding GAF image are symmetric.

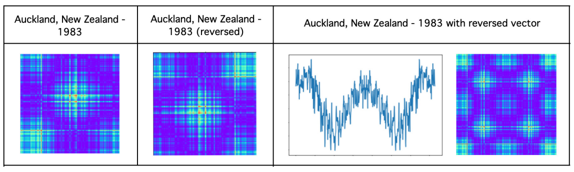

Figure 3: 'Same' class example: daily weather in Auckland (New Zealand) in 1983. Pairwise mirror vector and corresponding GAF image are symmetric.

4.2.2 'Different' Class: Coalesce Vectors with Reversed Second Vectors

To generate training data images for the 'different' class we will follow these steps:

- To randomly select different pairs of vectors we will shuffle first vectors and reverse second vectors
- The first vectors we will label as 'city1-year1' and second (reversed) vectors as 'city2-year2'
- Than we will concatenate vector pairs and label pairwise vectors as 'city1-year1-city2-year2'
- We will mark image class type as 'different'.

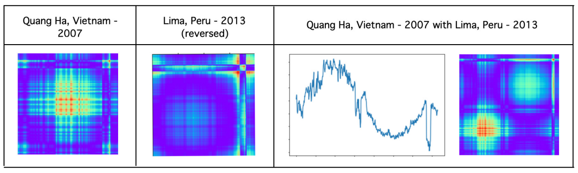

Figure 4: 'Different' class example: daily weather in Quang Ha (Vietnam) in 2007 and Lima (Peru) in 2013. Pairwise vector and corresponding GAF image are asymmetric.

Please see Figure 4 as an example of 'different' class images. This figure illustrates that concatenated different vectors and corresponding GAF image are asymmetric.

Image transformation code is described in fast.ai forum [15] and CNN classification code for fast.ai transfer learning method is described in [6].

4.3 CNN Image Classification

To estimate the results we will calculate accuracy metrics as the proportion of the total number of predictions that were correct. The training model accuracy metric was about 96.5 percent.

4.4 Entity Pairs with Two-way Relationships

GAF images are based on polar coordinates, therefore transforming pairwise vectors to GAF images might generate inconsistent results for turned around entity pairs.

To prove this hypothesis we will select 66 cities from continental West Europe with daily temperature data for 1992. For all city pairs we will create pairwise vectors in both directions, i.e. Paris - Berlin and Berlin - Paris. Then we will transform vectors to GAF images and run these images through the trained model to calculate their 'same' or 'different' probabilities.

As the results of using this model interpretation, most of city pairs in both directions have 'different' or 'same' metric. Out of the total number of city pairs (4290 city pairs) about 85 percent (3660 city pairs) are consistent. However about 15 percent of city pairs (630 city pairs) have inconsistent results. Please look at Table 1 for inconsistent city pair examples. Some of city pairs are located nearby, like Turin (Italy) and Monaco (Monaco) and some of city pairs are located far from each other, like Helsinki (Finland) and Bern (Switzerland).

4.5 Entity Pairs with One-way Relationships

We proved the hypothesis that pairwise vector classification model is not reliable for similarity prediction of pairs with two-way relationships and therefore this model should be used only for classification of entity pairs with one-way relationships.

Pairs of Cities	diff	same
Helsinki (Finland) – Bern (Switzerland)	0.41	0.59
Bern (Switzerland) – Helsinki (Finland)	0.74	0.26
Naples (Italy) – Lisbon (Portugal)	0.25	0.75
Lisbon (Portugal) – Naples (Italy)	0.92	0.08
Turin (Italy) – Monaco (Monaco)	0.53	0.47
Monaco (Monaco) – Turin (Italy)	0.18	0.82
Dresden (Germany) – Berlin (Germany)	0.16	0.84
Berlin (Germany) – Dresden (Germany)	0.50	0.50

Table 1: Examples of City Pairs with Inconsistent Same or Different Probabilities

Here we will show two scenarios of using this model to compare daily temperatures of pairs with one-way relationships. For the first scenario, we will calculate average vector of all yearly temperatures vectors for cities in Western Europe and compare it with yearly temperature vectors for all cities. For the second scenario we will determine a city located in the center of Western Europe and compare this city temperatures for years 2008 and 2016 with other cities.

4.5.1 Compare City, Year Temperature Vectors with Average of All Yearly Temperatures

To find cities with yearly temperatures similar to average temperatures for years from 1980 to 2019 in Western Europe we will calculate average time series for 2640 daily temperature time series (40 years and 66 cities).

As average of average temperature vectors provides a very smooth line, we don't expect that many city-year temperature vectors would be similar to it. In fact only 33 of city-year temperature time series have higher than 0.5 'same' probabilities and 22 pairs have probabilities higher than 0.65.

It is interesting that most of cities with high probabilities to be similar to average temperature are located on Mediterranean Sea, not far from each other. Here is a clockwise city list: Marseille (France), Nice (France), Monaco (Monaco), Genoa (Italy), Rome (Italy), Naples (Italy), and Salerno (Italy).

4.5.2 Compare City, Year Temperature Vectors with Central City

Here is another scenario of using pairwise vector model to compare daily temperatures of pairs with one-way relationships. From the Western Europe city list we selected the most centrally located city - Stuttgart (Germany) and created pairwise vectors by concatenating temperature vector pairs other city, Stuttgart for the years 2008 and 2016. Then we transformed pairwise vectors to GAF images and analyzed 'same' or 'different' probabilities by running images through our trained model.

Based on the model, for both years cities located close to Stuttgart had high probability of similar temperature vectors and cities located far from Stuttgart had high probabilities of different temperature vectors. Please see Figure 5 as an example of two cities that had very different temperatures in both 2008 and 2016:

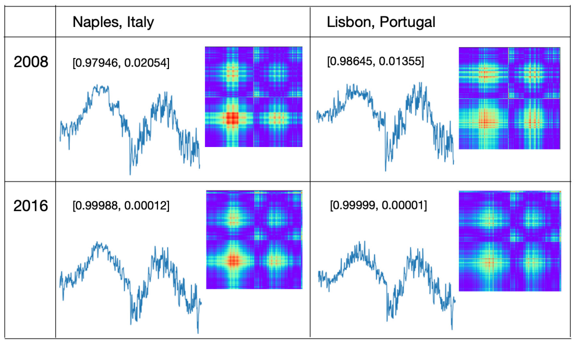

Figure 5: Both Naples (Italy) and Lisbon (Portugal) are located far from Stuttgart (Germany) and in both 2008 and 2016 years these cities had daily temperatures very different that daily temperatures in Stuttgart.

More difficult was to predict which cities had similarities with Stuttgart temperatures on the border between 'different' and 'same'. Please see Figure 6 with examples of such cities.

Notice that in both years Stockholm (Sweden) was on the 'same' side and Nice (France) was on 'different' side. So if we would use probability 0.5 as a threshold, we would consider Stockholm daily temperatures as similar and Nice daily temperatures as not similar to Stuttgart daily temperatures for years 2008 and 2016.

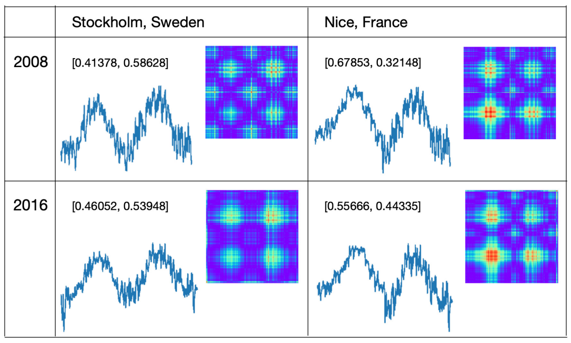

Figure 6: Two cities that were 'on the border' in years 2008 and 2016: Stockholm (Sweden) and Nice (France). Comparing with Stuttgart daily temperatures, both cities had probabilities of 'same' or 'different' close to 0.5.

5 Conclusion

In this study we introduced a novel unsupervised time series classification model. This model is embedding entities to vectors and combining entity pairs to pairwise vectors. Pairwise vectors are transformed to Gramian Angular Fields (GAF) images and GAF images are classified to symmetric or asymmetric classes using transfer learning CNN image classification.

We examined how this model works for entity pairs with two-way and one-way relationships and indicated that it is not reliable for classification of entity pairs with two-way relationships.

Based on climate data of cities from West Europe we demonstrated two successful scenarios for entity pairs with one-way relationships. In one scenario we selected centrally located city and found cities with similar and dissimilar daily temperature vectors. In another scenario we showed that European cities with the most smooth climate are located on Mediterranean Sea.

6 Broader Impact

There are several interesting directions in which this work could be extended. First, daily temperature time series data used in this study has spatial city location data. Projecting time series images to geographical locations will support fine-grained pattern detection. This practice is applicable to data mining scenarios where data that is a combination of spatial data and embedded vectors such as Electroencephalography pattern recognition [16].

Second, the time series classification approach of this study is implemented through transforming time series pairs to pairwise vectors and vectors to symmetric or asymmetric GAF images. In addition to time series, this process can be applied to a variety of embeddable entities such as words, documents, images, videos, etc. [14]. For example, this model can be used as unsupervised outlier detection in finding stock price time series that are very different from average stock prices.

Third, pairwise vectors model trained on symmetric and asymmetric GAF images can be train on some data domain and used for other data domains. For example, the model can be trained on time series data and used for word similarity classification.

Finally, the 'same' probability metric of pairwise vectors can be used to measure differences between vectors instead of measuring them typically through cosine similarities. Furthermore, through this metric direct graphs can be built for graph mining.

References

[1] S. Ardabili1, A. Mosavi, M. Dehghani, and A. R. Varkonyi-Koczy. Application of deep convolutional neural networks for detecting extreme weather in climate datasets. 2019. doi:10.20944/preprints201908.0166.v1.

[2] A. Bardes, J. Ponce, and Y. A. LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *ArXiv*, abs/2105.04906, 2021. doi:arXiv:2105.04906.

[3] J. Bromley, Y. LeCun, and other. Signature verification using a siamese time delay neural networ. *Journal of Pattern Recognition and Artificial Intelligence*, 7 (04):669–688, 1993. doi:10.1142/S0218001493000339.

[4] M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges, 2021. doi:10.48550/arXiv.2104.13478.

[5] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton. A simple framework for contrastive learning of visual representations. *ArXiv*, abs/2002.05709, 2020. doi:PMLR119:1597-1607.

[6] fast.ai. Practical deep learning for coders. https://course.fast.ai/, 2020.

[7] forum.fast.ai. Time series/ sequential data study group. https://forums.fast.ai/t/time-series-sequential-data-study-group/29686, 2019.

[8] J. Howard and S. Gugger. *Deep Learning for Coders with fastai and PyTorch.* O'Reilly Media, 2020. doi:978-1492045526.

[9] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4):917–963, 2019. doi:10.1007/s10618-019-00619-1.

[10] kaggle.com. Temperature history of 1000 cities 1980 to 2020. https://www.kaggle.com/hansukyang/temperature-history-of-1000-cities.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. doi:10.1145/3065386.

[12] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, vol. 521, no. 7553, pp. 436–444, 2015. doi:10.1038/nature14539.

[13] Y. Liu and E. Racah. Deep learning and machine learning in hydrological processes, climate change and earth. 2019. doi:10.20944/preprints201908.0166.v1.

[14] K. Nozawa. Something 2 vec. https://gist.github.com/nzw0301/333afc00bd508501268fa7bf40cafe4e.

[15] I. Oguiza. Time series - olive oil country. https://gist.github.com/oguiza/c9c373aec07b96047d1ba484f23b7b47, 2019.

[16] A. Romanova. Time series pattern discovery by deep learning and graph mining. In G. K. et al., editor, *Database and Expert Systems Applications - DEXA 2021 Workshops*, volume CCIS 1479, pages 192–201. Springer Nature Switzerland AG 2021, 2021. doi:10.1007/978-3-030-87101-7.

[17] sparklingdataocean.com. Free associations. http://sparklingdataocean.com/2019/06/01/word2vec2CNN/, 2019.

[18] sparklingdataocean.com. Cnn image classification for climate data. http://sparklingdataocean.com/2021/04/04/cityTempCNN/, 2021.

[19] sparklingdataocean.com. Unsupervised deep learning for climate data analysis. http://sparklingdataocean.com/2021/08/01/unsupervisdCityTempCNN/, 2021.

[20] Z. Wang and T. Oates. Encoding time series as images for visual inspection and classification using tiled convolutional neural networks. *Association for the Advancement of Artificial Intelligence*, 2015. doi:10.3389/fbuil.2021.660103.

[21] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. doi:10.48550/arXiv.1411.1792.