Self-supervised learning for time series

Individual Project I.

Author:
Borbély Bernárd
Applied Mathematics MSc

Supervisor:
Csiszárik Adrián
Alfréd Rényi Institute of Mathematics

Eötvös Loránd University
Faculty of Science
Budapest, 2023.

1. Self-supervised learning

In machine learning, it often happens that a large amount of unlabeled data is available, such as in medical diagnostics with EEG, ECG, EMG. Since regression and classification tasks heavily rely on existing labels, having a portion of the data unlabeled can make it challenging to utilize.
My goal for the semester was to get familiar with the basic principals of self-supervised learning and understand how is it realized in the domain of time-series and pre-training.
The goal of self-supervised learning is to leverage the information content of unlabeled datasets when solving a task, sometimes even using a different domain and then transferring the knowledge. This allows us to work with more data, particularly in the realm of time series where there is an abundance of unlabeled data, making self-supervised learning a valuable tool.
Typically working with unsupervised-learning involves creating a new task, based on the data thus transforming the unlabeled data into labeled data. One such example is when we mask out the end of the time-series and then the new labels can be the masked out data. Subsequently, supervised pretraining is performed using these new labels. Once this is completed, fine-tuning is carried out on the target task. For the new task the labels are devised in a way that we believe is necessary for the network to understand the data in order to solve the given problem.

> While self-supervised learning is flourishing in computer vision and natural language processing, it is still and open question whether this holds true in the time-series domain.

Throughout the semester our goal was to examine and better understand this issue.

2. Time Frequency Consistency framework

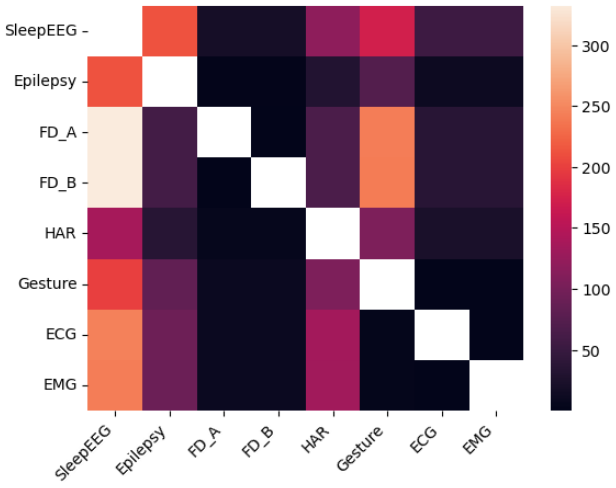
I dealt with contrastive pretraining on time series in my independent project, based on the framework proposed by Zhang, Zhao, and their colleagues [2]. I selected this approach, as it is a quite recent technique providing robust baseline for pre-training.
The essence of the method is to to create a representation of the data points in a in such a way that pairs of data considered similar (positive pairs) are close to each other, while those deemed different (negative pairs) are distant. For example in image classification positive pairs could be pairs of pictures containing the same object (a tea kettle), and negative pairs are every other picture (not containing a tea kettle).
It's not clear how to designate these positive and negative pairs, so for each data point we define a new one, which will be the positive pair. One way to create a positive pair for a data point is to slightly modify the time series, thus creating the positive pair. This could be done in the Fourier-space, so the change is a global. Another idea is to use the Fourier representation of the time-series, in essence a different representation of the data.
The main concept of the method is to use the Fourier-transform of the time series. Subsequently, embed both representation into the same time-frequency space (with two different functions), with the objective of making the two embeddings close to each other.

The issues of generalization with the TFC method To test the capabilities of the system, the original article works with 8 datasets. For the framework, 4 datasets were designated for pretraining and 4 for fine-tuning, allowing for testing various training setups.
In the first set of tests, they used 1 pretraining dataset and its corresponding pre-determined fine-tuning dataset. In the second setup, after using 1 pretraining dataset, tests were conducted on all 4 fine-tuning sets. Both setups yielded successful results, competing with and surpassing state-of-the-art models.
As a final inquiry, they examined a few test cases where pretraining was done by combining multiple pretraining datasets, followed by testing on individual fine-tuning datasets. Surprisingly, the method

yielded poorer results compared to the 1-1 setup. Moreover, *the more datasets were combined for pretraining, the worse the results became on the fine-tuning datasets.*
As this is a surprising result, we wanted to examine the reasons behind this phenomenon.

3. Experiments

Our initial hypothesis was that the similarity between the pretraining and fine-tuning datasets would serve as an indicator of how successful the fine-tuning would be.
To test this, we needed to define a metric between two datasets so that we could compare it with the performance provided by either loss or classification accuracy. Futhermore, the given framework [1] allowed only 4 dataset for pre-training, and 4 for fine-tuning, so we had to expand the scope of the system. The datasets were syntactically very different, with varying sampling frequencies, time series data points per pattern, and channel numbers.
Therefore, we conducted the comparison in Fourier space. First, we took all elements of the given dataset, performed Fourier transformation, averaged the results, and took the absolute value of the averaged elements.



1. ábra. Comparing the similarity of the dataset pairs. The columns are the pre-training dataset and the rows are the fine-tuning.

The length of the pretraining dataset served as the baseline, and we adjusted the transformed fine-tuning dataset accordingly. If the samples of the fine-tuning dataset were shorter, we padded the end with zeros. If the fine-tuning dataset was longer, we truncated the end of the averaged vector. Then, we computed the squared sum of the quadratic differences between these two vectors. The result can be seen in 1.figure. The columns correspond to the pre-training dataset and the rows are the fine-tuning. We can see that, there are several pre-training dataset which has a small distance from all of the fine-tuning ones. These would be: *FD_A*, *FD_B*, *ECG* and *EMG*. If our assumption is right, we would expect that pre-training on these datasets would yield the more succesful fine-tuning cases. If this is the case pre-training on *SleepEEG* and fine-tuning on any other dataset would result in poorer performance.

Future works In nature the Fourier transforms are such, that the small frequencies dominate and they are the important ones. If our test would show that the problem behind the poor preformance is the dissimilarity of the time-series datasets, one could implement augmentation methods to bridge over these differences. For example various mixup methods.

References

[1] Xiang Zhang és tsai. *Self-Supervised Contrastive Pre-Training For Time Series via Time-Frequency Consistency*. `https://github.com/mims-harvard/TFC-pretraining`. 2022.

[2] Xiang Zhang és tsai. "Self-Supervised Contrastive Pre-Training For Time Series via Time-Frequency Consistency". *Advances in Neural Information Processing Systems*. Szerk. S. Koyejo és tsai. 35. köt. Curran Associates, Inc., 2022, 3988–4003. old. URL: `https://proceedings.neurips.cc/paper_files/paper/2022/file/194b8dac525581c346e30a2cebe9a369-Paper-Conference.pdf`.