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 Introduction

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.

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 self-supervised learning involves creating a new task, based on the data thus transforming the unlabeled data into labeled data. An example of this would be masking out a portion of a time-series, then presenting a neural network with 9 possible choices to fill the masked out time-series part. 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 an open question whether this holds true in the time-series domain.

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

1.1 Time Frequency Consistency framework

I dealt with contrastive pre-training on time series for my independent project, based on the framework proposed by Zhang, Zhao, and their colleagues [1]. I selected this approach, as it is a quite recent technique providing robust baseline for pre-training.

The approach focuses on creating a pre-training framework on a large unlabeled dataset, and then subsequently fine-tune on a small, labeled dataset, utilizing transfer learning in the process.

The framework's goal is to create a representation (embedding) of the data points in such a way that similar data point's embeddings are close to each other, and different ones are distant.

However it is hard to tell which time-series pairs are similar, so we need a different approach, and this is where the framework utilizes the Fourier-transform of a time-series. The Fourier-transform belongs to the same underlying data, but represents that in a completely different way.

Let x_i^T be the time-series representation of the ith data point, and x_i^F be the Fourier-transform representation. The framework creates a z_i^T embedding from x_i^T and a z_i^F from x_i^F; into the same Time-Frequency domain and we set the goal z_i^T and z_i^F being close to each other. For the representation to not fall into a singularity we also want that for any other j data point z_i^T and z_j^T be far from each other. This is called contrastive learning, and it's quite popular in self-supervised learning. In contrastive learning x_i^T and x_i^F are called positive pairs, while x_i^T and x_j^T are called negative pairs. (x_i^T and x_j^F are also negative pairs.) In general we want positive pairs to be close to each other, and negative pairs to be far.

The issues of generalization with the TFC method To test the capabilities of the system, the original article worked with 8 datasets. For the framework, 4 datasets were designated for pre-training and 4 for fine-tuning, allowing for testing various training setups.

In the first set of tests, they used 1 pre-training dataset and its corresponding pre-determined fine-tuning dataset. In the second setup, after using 1 pre-training 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 pre-training was done by combining multiple pre-training 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 pre-training, 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.

2 Question

In the vision domain access to more data resulted in higher performance, even when the additional data came from unrelated fields to the original task. For example pre-training on ImageNet where the pictures contain everyday objects, helped solve chest x-ray classification tasks. The same could be said for the language domain, where training often involves all kind of digitalized documents from a huge range of topics. In both of these domains however there is a strong common underlying structure which could be an explanation for why using larger datasets yields better results. In the vision domain each picture can be boiled down to edges and basic shapes which make up the whole picture. In the language domain the grammatical rules and regularities can provide a similarity between completely different topics.

The many-to-one setup fail of the Time-Frequency Consistency framework could mean that in the time-series domain there is no such common underlying structure, and the time-series domain might be too heterogeneous. For example we might combine datasets containing boring machines' behaviour and EEG medical devices and the sampling rate, length could differ with patterns recurring with different rates.

We aimed to understand how different pre-training dataset compositions work together and affect each other.

3 Experimental Setup

We have 4 datasets dedicated for pre-training (SleepEEG, FD-A, HAR and ECG) and 1 for fine-tuning (Epilepsy).

To examine how different pre-training dataset compositions affects each other, first we created the baseline where we measured the performance for each dataset using only that, as a pre-training dataset. These were the 1-to-1 experiments. After this, we created every combination of the dataset pairs (6), these were the 2-to-1 experiments. Then we tested every triplet combination (4), the 3-to-1 setup and finally, the 4-to-1 setup, where we combined all of them.

Additionally we tried different augmentation techniques such as mixup and adding the fine-tuning dataset to the pre-training datasets.

Mixup is a common data augmentation technique in deep learning to bridge the differences between 2 datasets. Mixup generates a weighted interpolation of two data points from the training data. Given (x_i, y_i) and (x_j, y_j) where x_i are data points with their corresponding label y_i, mixup creates $(\hat{x}, \hat{y})$:

$$\hat{x} = \lambda * x_i + (1 - \lambda) * x_j$$

$$\hat{x} = \lambda * y_i + (1 - \lambda) * y_j,$$

where $\lambda \in Beta(\alpha)$, and α is a fixed parameter, for the distribution.

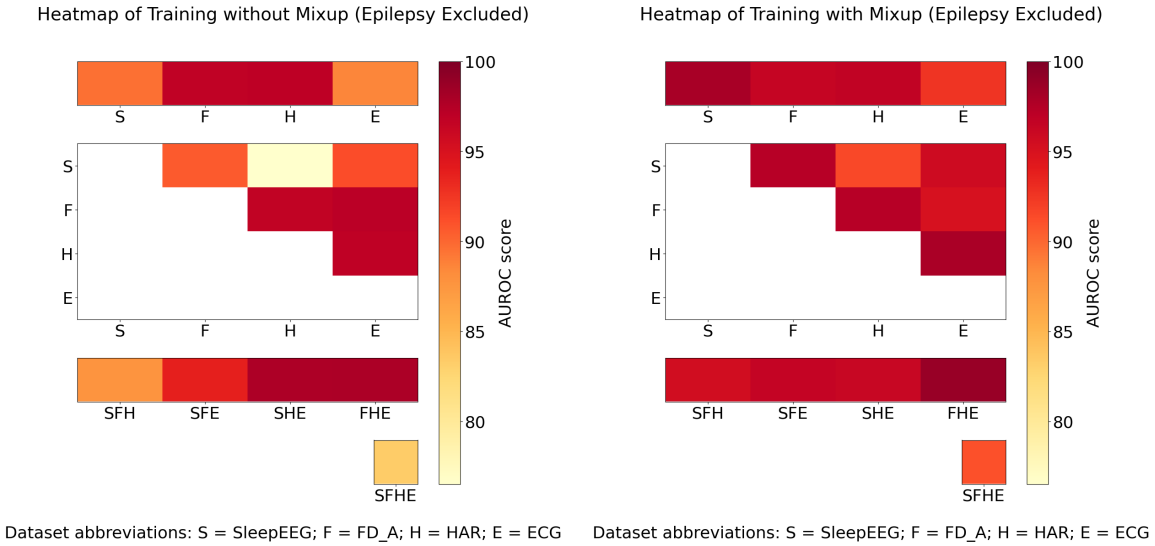
We also observed that the different datasets contain different amount of training data, while SleepEEG has ≈ 370000 samples, FD-A contains only ≈ 8000. As the number of samples seen by the network can greatly affect the network's ability to generalize, as a final inquiry we tried fixing the number of samples passing through the network.

Using mixup, adding Epilepsy to the pre-training dataset and fixing the number or samples used were independent parameters, which could be turned on or off independently from one another.

Every parameter mentioned belongs to the pre-training model. The fine-tuning on Epilepsy is a binary classification task. For each pre-training model we tested the embedding quality with the same classifier with 3 different fine-tuning seeds. The performance of the pre-training model is characterized by the classifier's AUC score.

4 Results

Examining 1a (no mixup, Epilepsy excluded) alone we can observe that using only 1 dataset for pre-training offered a strong baseline. Then during the 2 dataset experiments the performance dropped only when SleepEEG (abbreviated as S on the figure) was involved. After that combining any 3 dataset achieved top performance, which finally dropped below baseline, when the 4 datasets were combined.



(a) No mixup, Epilepsy excluded from pre-train

(b) Using mixup, Epilepsy excluded from pre-train

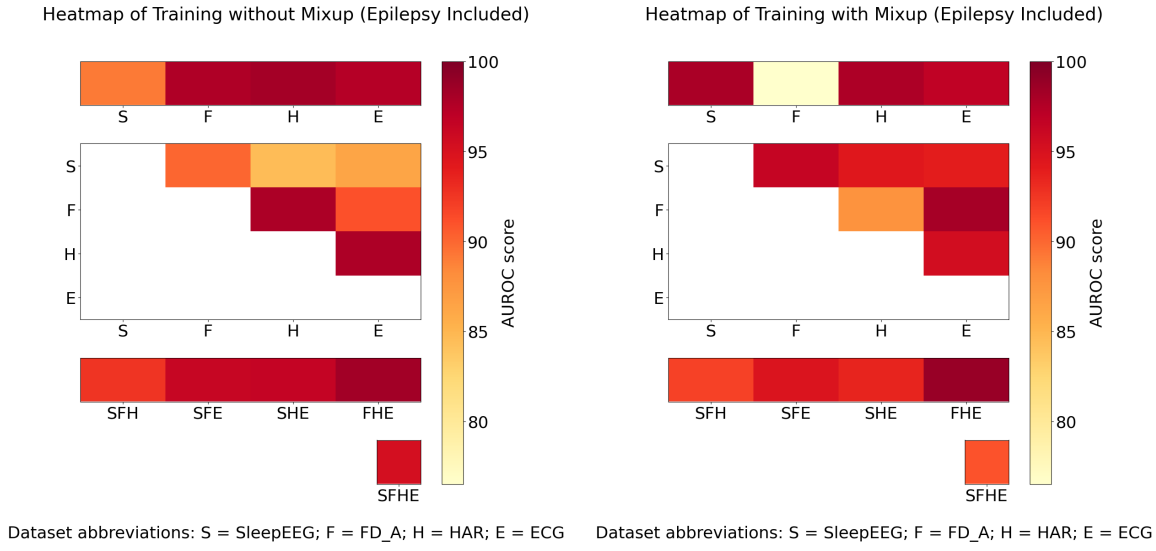
Figure 1: No matter the pre-train dataset composition, using mixup yields success.

Each figure represents a training setup we wanted to examine. The number of training samples seen was fixed and each network has seen the same amount. The difference comes from whether we are using mixup, and whether we are adding the target dataset to the pre-training. The scales are fixed, so the figures are directly comparable to each other. The performance was measure with $AUROC$ score.

A figure contains 4 major subplots corresponding to different pre-training dataset setups. The first heat map (a row) belongs to the case when there is only 1 pre-training dataset, and the

label underneath a square corresponds to the dataset used. The second subplot belongs to the possible dataset pair combinations. Because the order of the merging doesn't matter, only the upper triangle matrix is shown. Each dataset's abbreviation appears in the rows and the columns as well. Taking the intersection corresponds to using that 2 dataset for pre-training. For example taking the intersection of row S and column E corresponds to using SleepEEG and ECG for pre-training. The third subplots belong to the possible dataset triplets, and the label underneath the boxes were created from the dataset abbreviations which are present in the given pre-training setup. For example FHE corresponds to using FD-A, HAR and ECG for pre-training. The last subplot is the case when all datsets are used for pre-training.

Comparing Figure 1a and Figure 1b we can read that using mixup seems to stabilizes and improve performance in virtually every case. This suggests that mixup contributes to higher $AUROC$ score.



(a) No mixup, Epilepsy included in pre-train

(b) Using mixup, Epilepsy included in pre-train

Figure 2: Adding the target dataset to pre-training.

Comparing Figure 2a and Figure 2b we see that using mixup not necessarily meant higher performance, but neither did it mean lower.

Comparing Figure 1a (no mixup, Epilepsy Excluded) and Figure 2a (no mixup, Epilepsy Included), we can also observe a clear improvement and stabilization in performance.

However when we take a look at 2a (using mixup, Excluding Epilepsy) and 2b (using both), we can observe that excluding Epilepsy almost seems to be beneficial to the performance.

5 Discussion

During this semester we aimed to better understand how different pre-training dataset compositions work together and affect each other. We tried multiple augmentation techniques, which brought promising results, but some additional work might be needed to achieve maximal affect with these techniques. We also observed that combining 2 datasets is the trickiest, where the augmentation techniques helped the most. The combination of 3 datasets already seemed robust, where there

wasn't a lot of room for improving with augmentations. However combining all 4 still results in poorer performance, even with different and multiple augmentation techniques.

Looking forward, our research could benefit from exploring additional augmentation techniques while continuing to refine the ones currently in use. Repeating these experiments would provide a more comprehensive understanding. Evaluating the dataset compositions on a similarity measure between the pre-training and fine-tuning datasets could offer valuable insights into dataset heterogeneity.

In conclusion we identified promising starting points for further explanation. By continuing to refine our approaches and expanding our experimental scope, we aim to gain a deeper understanding of the time-series domain in machine learning. Our goal is to develop more effective strategies for leveraging diverse datasets, ultimately improving model performance and robustness.

References

[1] Xiang Zhang et al. "Self-Supervised Contrastive Pre-Training For Time Series via Time-Frequency Consistency". In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo et al. Vol. 35. Curran Associates, Inc., 2022, pp. 3988–4003. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/194b8dac525581c346e30a2cebe9a369-Paper-Conference.pdf.