SELF-SUPERVISED REPRESENTATION LEARNING FROM ELECTROENCEPHALOGRAPHY SIGNALS

IEEE 29th International Workshop on Machine Learning for Signal Processing, 2019

Presenter: Nooshin Taheri

7/31/2024

Introduction:

- The impressive success of deep learning in various domains can, in large part, be explained by the availability of large labeled datasets.

- Challenges of labeling EEG signals:

  - Accurate physiological data annotations can be costly, time-consuming, or impossible.
  - Noise and complexity make EEG signal interpretation difficult.
  - Understanding participants' thoughts or actions in experiments is challenging, hindering accurate labels.

- Self-supervised learning (SSL) is an unsupervised learning approach that learns representations from unlabeled data, exploiting the structure of the data to provide supervision.

Self-Supervised Learning (SSL):

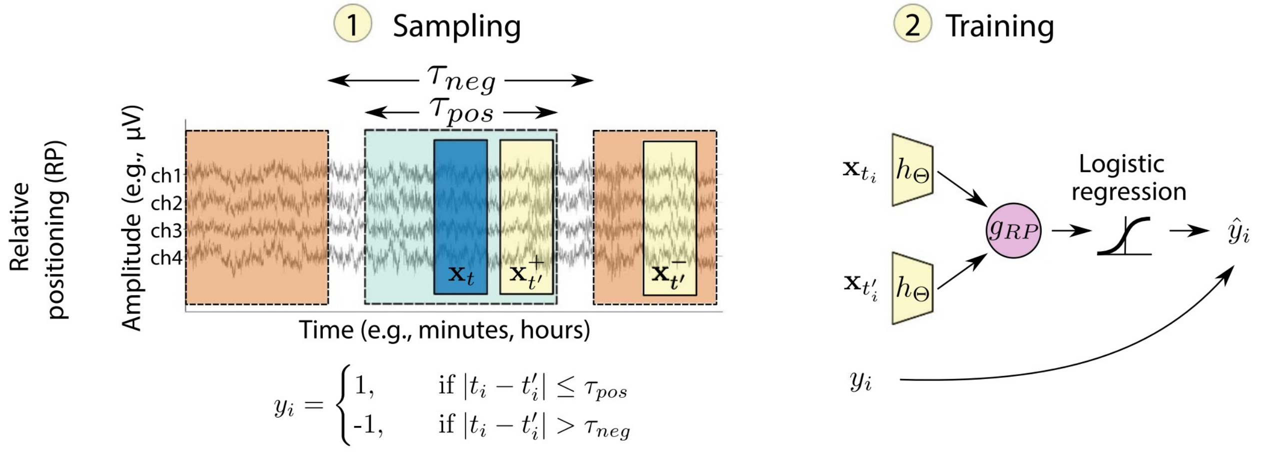
- SSL comprises a ‘pretext’ and a ‘downstream’ task.
- The downstream task is the main task of interest, often with limited or no annotations.
- The pretext task must be related to the downstream task to use similar representations.
- In the pretext task, annotations are generated using only the unlabeled data to capture the general representation of the data.
- Apart from facilitating the downstream task and/or reducing the number of annotated examples necessary, self-supervision can also uncover more general and robust features than those learned in a specialized supervised task.
- This paper introduces two temporal contrastive learning tasks that we refer to as “relative positioning” and “temporal shuffling”.

Self-supervised learning pretext tasks for EEG

- Relative positioning (RP):
  - The relative positioning task is designed to learn the structure of EEG data by comparing pairs of time windows (segments of EEG data).
- The EEG data is divided into smaller segments called time windows with a fixed duration T.
- N pairs of windows from the EEG data are created and labeled:
- The first window x_t is referred to as the 'anchor window'.

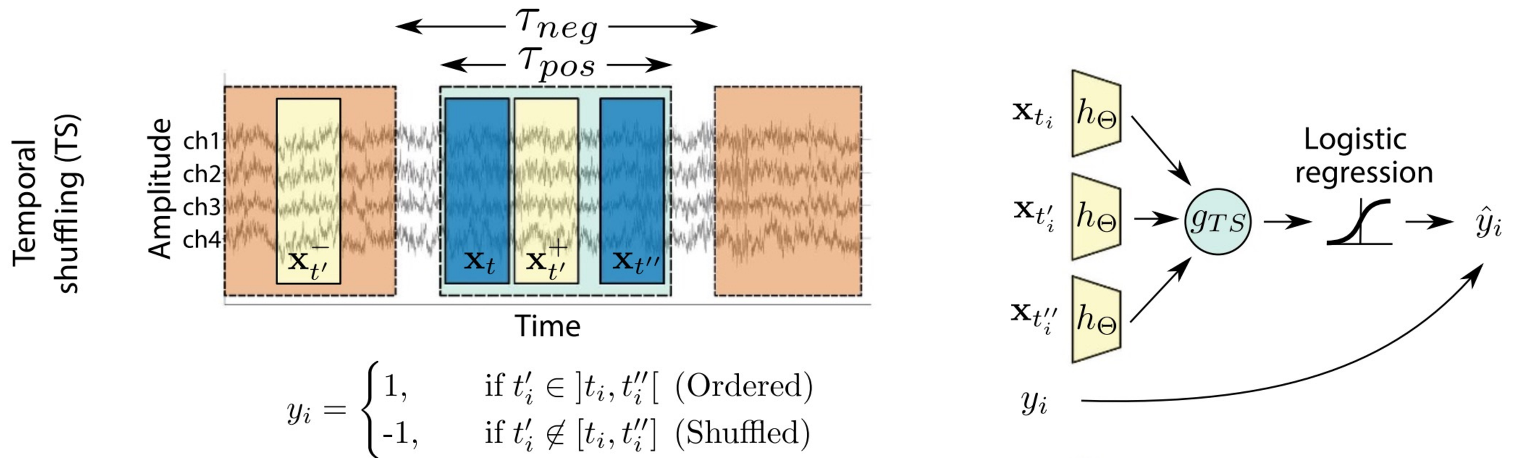
$$y_i = \begin{cases} 1 & \text{if } |t_i - t'_i| \leq \tau_{pos} \\ -1 & \text{if } |t_i - t'_i| > \tau_{neg} \end{cases}$$

- A feature extractor $h\Theta$ is applied to each window x_t and $x_{t'}$ to obtain their feature representations.
- The contrastive module gRP calculates the element-wise absolute difference between the feature representations of the two windows.
- ignore window pairs where $x_{t'}$ falls outside of the positive and negative contexts of the anchor window x_t.



Self-supervised learning pretext tasks for EEG

- Temporal shuffling(TP):
  - aimed at learning representations of EEG data by examining the order of time windows.
- In ‘temporal shuffling’ (TS), they sample two anchor windows , x_t and $x_{t''}$ from the positive context, and a third window $x_{t'}$ that is either between the first two windows or in the negative context.
- construct window triplets that are either temporally ordered ($t < t' < t''$) or shuffled ($t < t'' < t'$ or $t' < t < t''$).
- augment the number of possible triplets by also considering the mirror image of the previous triplets, e.g. (x_t, $x_{t'}$, $x_{t''}$) becomes ($x_{t''}$, $x_{t'}$, x_t)
- The label y_i then indicates whether the three windows are ordered or have been shuffled.



Dataset and Data preprocessing:

Datasets Used

Dataset	Number of Subjects	Age Range	Sampling Frequency	EEG Channels	Labeling	Labels (Stages)
Physionet Sleep EDF	83	25 to 101	100 Hz	Fpz-Cz, Pz-Oz	Trained technicians (R&K definition)	W, N1, N2, N3, R
MASS Dataset Session 3	62	20 to 69	256 Hz	Fz, Cz, Oz	AASM manual	W, N1, N2, N3, R

Data Preprocessing Steps

Step	Description
Filtering	4th-order FIR lowpass filter with 30-Hz cutoff frequency and Hamming window
Downsampling	MASS recordings downsampled to 128 Hz
Windowing	Non-overlapping 30-second windows (3000 samples for Sleep EDF, 3840 samples for MASS)
Normalization	Windows normalized to zero mean and unit standard deviation per channel
Sampling	2000 anchor windows uniformly sampled per recording; three positive and three negative tuples per anchor window based on temporal proximity

Model Architecture for the feature extractor h

- 1D-CNN:
  - Two convolutional layers
  - Two max-pool layers
  - A flattening layer

- input shape: (C,T,1)
  - C: the number of EEG channels.
  - T: The number of data points for each segment

- Activation function: Relu

- Optimizer:
  - Adam optimizer with $\beta 1$=0.9, $\beta 2$=0.999 and learning rate 0.001

- Number of epochs: 300 epochs, or until the validation loss does not decrease anymore for a period of at least 30 epochs.
- Batch size: 256
- Dropout: 50%

Compared Models

Baseline models	Description
Random Initialization	Model with random weights, not trained
Convolutional Autoencoder (AE)	Encoder-decoder architecture, trained to reconstruct input: • The encoder is similar to the feature extractor h used in the SSL tasks. • The decoder has four convolutional layers and aims to reconstruct the input data. • The model is trained using mean squared error as the reconstruction loss.
Purely Supervised Learning	This model uses the feature extractor h from the SSL tasks. An additional softmax layer is added to classify labeled data into one of the five sleep stages.
Human-Engineered Features	Traditional statistical and spectral features • **Statistical features**: mean, variance, skewness, kurtosis, standard deviation, • **Spectral features**: frequency log-power bands between (0.5, 4, 8, 13, 30, 49) Hz, Ratios of Frequency bands, peak-to-peak amplitude, Hurst exponent, approximate entropy and Hjorth complexity. • This results in 34 features per EEG channel, which are concatenated into a single vector.

By using balanced accuracy (the average per class recall) for evaluation and weighted loss during training, the authors ensure that their model performs well across all classes.

Experiments

Experiment 1: SSL models learn representations of EEG signals and facilitate sleep staging

- **Objective**: evaluate the ability of the CNN to learn the SSL tasks
- Method:
  - Train the feature extractor h on the entire training set using the SSL tasks (Relative Positioning (RP) and Temporal Shuffling (TS)) with three sets of hyperparameters τpos and τneg.
  - Project labeled samples into the feature space learned by the SSL tasks.
  - Train a multinomial linear logistic regression model on these features to predict sleep stages.

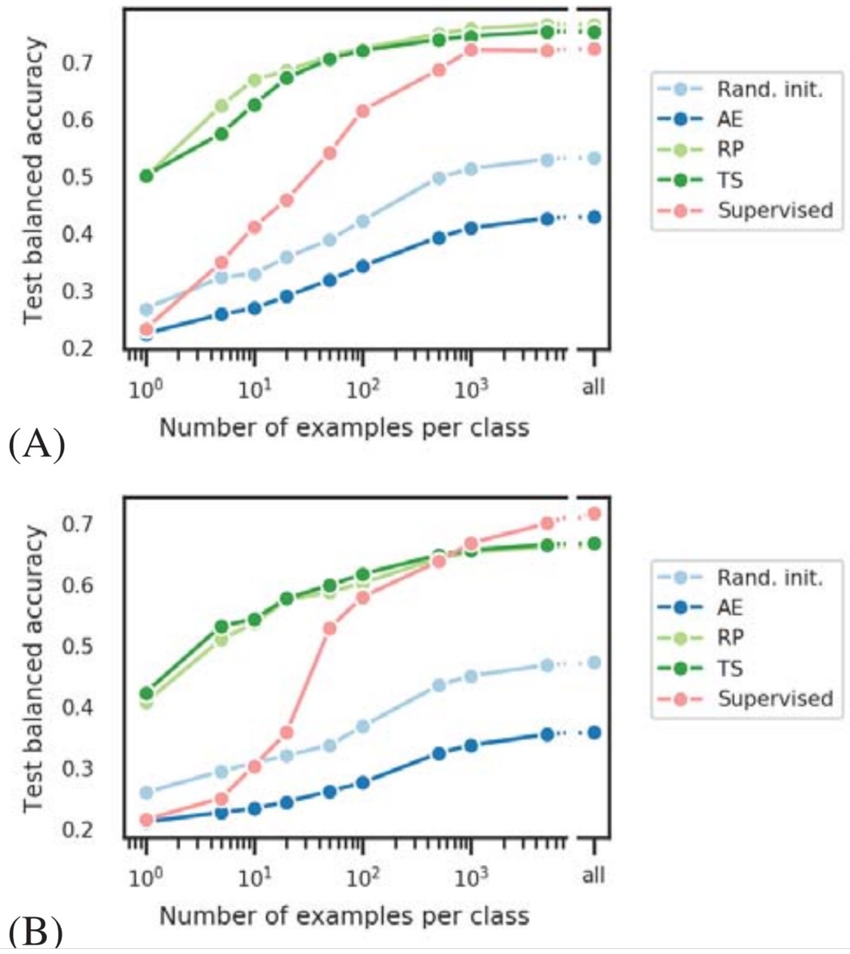
	τ_{pos}	τ_{neg}	bal acc$_{SSL}$	bal acc$_{staging}$
RP	2	2	**79.49**	75.73
	4	15	78.60	**76.66**
	120	120	56.30	65.71
TS	2	2	81.42	**75.90**
	4	15	**82.12**	75.37
	120	120	66.59	66.66
EEG features	-	-	-	79.43
Fully supervised	-	-	-	72.51

- Conclusion:
- SSL models achieved performance close to handcrafted features and significantly better than purely supervised models
- The SSL tasks successfully learned meaningful features from the EEG data, which facilitated sleep staging.

Experiments

Experiment 2: SSL enables sleep staging with limited annotated data

- **Objective**: evaluate SSL tasks when only a small amount of labeled data is available.
- Method:
  - Train different feature extractors using various approaches:
    - AE, RP and TS on unlabeled data; full supervision on labeled data, randomly initialized models
  - Use the trained feature extractors to obtain feature vectors from the EEG data.
  - Evaluate sleep staging performance using linear logistic regression models.
- Results:
  - Mass Data:
    - SSL features (RP and TS) significantly outperform the purely supervised model for all data regimes.
    - RP and TS both perform well, with RP slightly outperforming TS by a fraction of a percent in most cases.
    - Both AE and randomly initialized models perform much worse.
  - Sleep EDF Data:
    - Results are similar to the MASS dataset except:
    - When more than 500 labeled examples per class are available, the purely supervised model outperforms the SSL features.
    - TS performs slightly better than RP on this dataset.



(A) on MASS and **(B)** on Sleep EDF

Conclusion:

- The proposed SSL tasks, Relative Positioning (RP) and Temporal Shuffling (TS), effectively train feature extractors that capture meaningful EEG signal representations.
- Both proposed SSL tasks perform well, but RP is slightly more computationally efficient due to using only two windows instead of three.
- By reducing the amount of labeled data required to reach high performance, SSL tasks are promising alternatives to an expensive and time-consuming labeling process.