UNDER REVIEW

RESCIENCE C

Replication

[Re] Deep Convolution Neural Network and Autoencoders-Based Unsupervised Feature Learning of EEG Signals

Bruno Aristimunha[1], Diogo Eduardo Lima Alves[1], Walter Hugo Lopez Pinaya[1, 2] Raphael Y. de Camargo[1]

[1]Center for Mathematics, Computation and Cognition (CMCC), Federal Univesity of ABC (UFABC), Rua Arcturus, 03. Jardim Antares, São Bernardo do Campo, CEP 09606-070, SP, Brazil. – [2]Department of Psychosis Studies, Institute of Psychiatry, Psychology & Neuroscience, King's College London, London, UK

Edited by
(Editor)

Reviewed by
(Reviewer 1)
(Reviewer 2)

Received
–

Published
–

DOI
–

This paper presents our efforts to reproduce the results achieved by the authors of the original article [1]. We follow the steps and models described in their article and the same public data sets of EEG Signals. Epilepsy affects more than 65 million people globally, and EEG Signals are critical to analyze and recognize epilepsy. Although the efforts in the last years, it is still challenging to extract useful information from these signals and select useful features in a diagnostic application. We construct a deep convolution network and autoencoders-based model (AE-CDNN) in order to perform unsupervised feature learning. We use the AE-CDNN to extract the features of the available data sets, and then we use some common classifiers to classify the features. The results obtained demonstrate that the proposed AE-CDNN outperforms the traditional feature extraction based classification techniques by achieving better accuracy of classification.

1 Introduction

The paper *Deep Convolution Neural Network and Autoencoders-Based Unsupervised Feature Learning of EEG Signals* [1] presents an unsupervised approach to learn from EEG signals of epilepsy patients. It uses dimension reduction algorithms to extract features since the original data is high-dimensional. Then, it uses different common classifiers to classify the features obtained.

We re-implement the author's method from scratch as best as we could by using only the paper as instruction. In addition, we get new results by combining the proposal classifiers into a classifier by set voting.

This paper is organized as follows: Section 2 introduces the methodological proposal employed, and their differences from [1]. Section 3 lists the experimental validation process using epilepsy datasets. Section 4 presents the corresponding results and analyzes our approach. Finally, conclusions were summarized in Section 5.

2 Methodology Proposal

In this section, we describe implementation details, as the core is the reproducible aspect of our reference article. We introduce the idea and implementation of autoencoder/feature learning and our version of the model in [1], explaining the differences we have made in the original model.

Correspondence should be addressed to Bruno Aristimunha (b.aristimunha@gmail.com)
The authors have declared that no competing interests exists.
Code is available at https://github.com/bruAristimunha/ReScience-submission.

2.1 Implementation Details

We decided to reproduce the implementation described in the article using Keras [2] with TensorFlow [3] as backend. Our repository includes a list of all the required libraries employed in acquiring the datasets and running the model (the original and the proposed one). Using to the methodology proposed in [4], we store all the checkpoints for the trained models for reproduction purposes. Besides that, the training logs can be visualized using the ensorBoard tool.
Given the lack of information about some implementation details in the original paper, some assumptions or cuts are made:

- The number of epoch in the AutoEnconder is assumed to be 5000;
- The number of samples per batch size is assumed to be 256;
- The last column of the Bonn University EEG dataset was removed since the authors of [1] used 4096 features and in this database there are 4097 column.
- In the Children's Hospital of Boston EEG database we use the channel reported by the author to train the AutoEncoder;
- The loss function presented in equation 12 in [1] was implemented;
- The value of the seeds used in all classifiers, data splitting and elsewhere was 42;
- The train-validation ratio was $80\% - 20\%$ for the to AutoEnconder. For the classifiers we used 5 or 10-fold cross-validation
- We resized the values using the MinMax method, before the classification process.
- The classifier presented in the final subsection (NN2) was not reproduced for lack of information;

We performed the experiments using a computer with an Intel Core i7-5930K at 3.50 GHz CPU and two GPUs: Nvidia Quadro K5200 and GeForce GTX 970. We also executed some experiments using a Nvidia Titan X GPU.

2.2 AutoEncoders

The autoencoder implemented is a specific case of neural network structure. It is formed by set three layers, an encoder layer, an decoder layer and a hidden layer. The training is done to set the weights of the hidden layer to force the input layer and output layer to be as close to each other as possible. Our features are extracted from the hidden layer, which reduces the dimension of data. For more details about the encoding and decoding functions see the Section 2 in the original article [1].

2.3 Feature Learning Model

In this subsection, we will omit equations and minor details (for complete information, see [5, 6]. Since we have the dimension reduced by autoencoder we focus on the next challenge: how to obtain effective features from the EG signals. The AE-CDNN implemented follows the steps:

- Encoder: sample input, convolution layer, down-sampling layer, reshape operation, full connection layer, and eature coding.
- Decoder: feature coding as input, full connection layer, reshape operation, deconvolution layer, up-sampling layer and sample reconstruction

Basically, the convolution layer acts as our feature extractor. It performs many successive convolution calculations overthe input data and allowing the extraction of useful features from the data. The pooling layer is a down-sampling method which reduces data dimension. In the convolution and pooling layer are uses windows to slide and extract the feature maps. These intervals do not overlap each other, and with then we obtain the pooled feature maps. The feature sizes tested were $m \in \{2, 4, 8, 16, 32, 64, 128, \}$. The convolution and pooling operations can be iterated multiple times. Reshape operation uses the pooled feature maps to construct an one-dimension vector and a full-connection layer to transform this one-dimension vector.
Considering x as the input and y as the output, now we need to re-transform the one-dimension vector which will generate the y output, recall we want to minimize the difference between x and y and we have the following equation to calculate loss Mean Absolute Error:

$$\text{Loss MAE} = \frac{1}{N}\sum_{i=1}^{N}|x^{(i)} - y^{(i)}|.$$

In addition, we have also used/implemented one more loss function, namely Mean Absolute Average Error - MAAE. The formula presented in the original article by [1] differs from the recognized Mean Absolute Percentage Error - MAPE formula, despite having similar intuitions. Thus, we chose to implement this loss equation, and we have not found its use elsewhere, the difference between the loss functions is only in the fact that one takes in the denominator the value per $x^{(i)}$ and the other takes the average $\bar{x}^{(i)}$, that are contained below:

$$\text{Loss MAAE} = \frac{1}{N}\sum_{i=1}^{N}\frac{|x^{(i)} - y^{(i)}|}{\bar{x}^{(i)}}.$$

Note that we refer to [1]'s AE-CDNN-L1 as *Loss-MAE* and to [1]'s AE-CDNN-L2 as *Loss-MAAE*

2.4 Classification

Since we have extracted the features with reduced dimension, we use supervised learning models on these features in order to classify the EEG signals. We evaluate each classifier and then we compare the results obtained with each one. The classical classifiers used are: K-Nearest Neighbors (K-NN), Support-Vector Machine - Linear Kernel and Radial Basis Kernel (SVM1, SVM2), Decision Tree (DT), Random Forest (RT), Multilayer Neural Network (MLP), Adaptive Boosting Algorithm (ADB) and Gaussian Naive Bayesian (GNB).

3 Experimental Methodology

In this paper, as in our reference paper [1], we use unsupervised learning method in EEG signals in order to obtain useful features. This process is needed because the original data is high-dimensional. By using the auto-encoder, we can extract features with reduced dimension. As the original authors we may refer to Bonn University EEG database simply as dataset 1 and to Children's Hospital of Boston EEG database simply as dataset 2.

3.1 Bonn University EEG database

We can use different approaches to detect epileptic crisis. Then, to obtain a comparative measure, we verify our outputs using the method described in 2 and the original

one showed in Section [1]. This database is public and was published by [7]. The study groups were the control, inter-ictal and ictal distributed into five sets (denotated A-E). Each containing 100 records of 23.6 seconds duration and frequency of 173.6 Hz on a single channel, with 12-bit resolution. Each data segment has 4097 samples. These recordings underwent a pre-processing in which the signals had a band filter between 0.53 to 40 Hz. There was also the removal of artifacts such as muscle movements or flicker movements.
We used labels A, B, C, D and E for the subsets. Set A corresponds to open-eye activity and subset B to closed-eye activity of 5 healthy volunteers The subsets C and D have interictal epileptiform activity from 5 epileptic patients. And E contains signals during epileptic patients' seizure (ictal intervals). According to [8], this dataset is a compilation of recordings under different conditions.

3.2 Children's Hospital of Boston EEG database

The second database, also public, contains the EEG signals from a Children's Hospital of Boston [5]. It was recorded by measuring the brain's electrical activity to obtain EEG signals by connecting multiple electrodes to the patients' scalp. The data incorporates the EEG signals of 23 children with refractory epilepsy.
This database, built in partnership with the Massachusetts Institute of Technology (MIT), has 5 men and 18 women between 3 and 22 years. The frequency range was 256 Hz with 16 resolution bits. Most patients contain 23 channels and some with 24 channels. In contrast to the first set of data, we have multiple channels here, then we need to select channels. The selection followed the methodology used in [9], which analyzes the variance of each patient, and after that, chooses the channel of greater variance to represent that individual. The channel reported by the authors was FT9-FT10.
In the data of the first ten patients, we chose two hundred 200 windows with a size of 4096 from the epileptic seizures and and another two hundred 200 when there are no epileptic seizures.

4 Results and Discussion

In this section, we present our reproduction results. In the first subsection we analyze the variance present in the channels. The second contains the reproduction of all possible tables and figures, with a discussion of the reasons for the differences.

4.1 Checking the Variance

According to the original authors, the choice of the channel in the Children's Hospital of Boston EEG database observed the variance present in the channels. For that, they followed the methodology: 1) calculate the variance of each channel in each sample, and select the channel with the maximum variance for each sample; 2) count the number of times each channel was selected; 3) select the channel the highest count among the first 10 patients.
Following the methodology presented above we found a different channel from the one found in [1]. So we decided to explore other scenarios. Note that the variance can be calculated based on each sample, each person or the full database. Despite the fact the authors in [1] mentioned explicitly they used the variance of each sample, since we found a different result we explored the other possibilities (per person and the full database).
Thereby, we modeled the three scenarios. In the first, we analyzed each recording file of the dataset as a sample, having an average length of 921600 points referring to the

recorded $3600s$. For each file we computed and selectedthe electrode with more variance. We counted the number of times each electrode was selected.The results obtained can be seen in the Figure below:
In the second scenario, we understand that each sample is accumulated per person with all his recordings. So the variance was calculated in parallel in the files and combined for each person. For each person, we count the occurrence of the channel with more variance. As shown in Figure 1b.
Finally, as a final scenario, we calculate the cumulative variance across all people and all records, thus, we did not perform a sampling process. In other words, we put all the files together and calculate the variance as if it were a single record. For this, we compute the variance, number of points and average per channel in each file and accumulate through the cumulative variance calculation algorithm. The result of this analysis approach can be seen in Figure 1c.
The results obtained in the first, second and third scenarios were not consistent with those reported by the author, in any of them we found the channel FT9-FT10 as the one appearing most for the first 10 patients. There is also another possible scenario, however, it is not reproducible, where the authors randomly sampled the dataset and evaluated the variance.
Given the associated uncertainty, we decided to repeat the choice of the channel $FT9 - FT10$, although this is not the one with the most variance in the modeled scenarios.

4.2 Reproduction of the values reported by the original author

The results obtained in our reproduction experiment, for the first dataset, are presented in Accuracy Tables 1 and 2 and the differences between results can be seen in Figures 2 and 3. We employed the same methodology as the one used in the original paper, performing a 5-fold cross-validation for each classifier, and we show the mean values. In the first column, we have the encoded hidden layer size[1] after the feature learning process. The remaining columns are the average values obtained in the classification using the latent space as input. For each table reproduced, we also present the original result and the difference between them.

Dimension	k_neighbors	svm_linear	svm_radial	decision_tree	random_forest	multi_layer	ada_boost	gaussian_nb	average
2	0.480	0.600	0.600	0.600	0.600	0.600	0.600	0.400	0.56000
4	0.708	0.600	0.600	0.688	0.682	0.600	0.738	0.560	0.64700
8	0.764	0.600	0.600	0.728	0.756	0.600	0.710	0.556	0.66425
16	0.786	0.600	0.600	0.762	0.778	0.586	0.766	0.588	0.68325
32	0.794	0.600	0.598	0.722	0.812	0.660	0.806	0.638	0.70375
64	0.814	0.602	0.696	0.722	0.786	0.718	0.796	0.650	0.72300
128	0.780	0.716	0.754	0.700	0.786	0.792	0.782	0.616	0.74075

Table 1. Classification results of MAE for dataset 1 - Table 2 in [11].

Dimension	k_neighbors	svm_linear	svm_radial	decision_tree	random_forest	multi_layer	ada_boost	gaussian_nb	average
2	0.738	0.600	0.60	0.694	0.694	0.600	0.730	0.450	0.63825
4	0.738	0.600	0.60	0.728	0.724	0.600	0.736	0.520	0.65575
8	0.712	0.600	0.60	0.694	0.658	0.600	0.742	0.562	0.64600
16	0.818	0.600	0.60	0.792	0.790	0.568	0.792	0.654	0.70175
32	0.796	0.600	0.60	0.724	0.786	0.602	0.804	0.644	0.69450
64	0.812	0.600	0.60	0.730	0.804	0.688	0.786	0.664	0.71050
128	0.790	0.616	0.72	0.726	0.758	0.750	0.772	0.620	0.71900

Table 2. Classification results of MAAE for dataset 1 - Table 3 in [11].

We can perceive some differences when compared to the original results. In Table 1, we obtained the best average with a dimension equal to 64, while the original document obtained the best average when the dimension is equal to 128. The original document obtained higher accuracy values in most cases, even though when the dimension is equal to 2 or 4, our accuracy values are higher. The best precision in our article and in the original article was obtained by the random forest algorithm.

[1] Denoted here as Dimension and is equivalent to the variable m in the original paper.

Dimension	k_neighbors	svm_linear	svm_radial	decision_tree	random_forest	multi_layer	ada_boost	gaussian_nb	average
2	0.7175	0.5025	0.5575	0.7400	0.6725	0.4525	0.7525	0.7700	0.645625
4	0.7700	0.5675	0.6875	0.7950	0.7525	0.7150	0.7925	0.6975	0.722187
8	0.8050	0.6050	0.7575	0.8100	0.8300	0.7650	0.8200	0.7800	0.771563
16	0.7950	0.7725	0.8600	0.7900	0.8450	0.8250	0.8200	0.8625	0.821250
32	0.7425	0.8300	0.8550	0.8325	0.8325	0.8500	0.8375	0.8625	0.830313
64	0.7650	0.8200	0.8400	0.7900	0.8375	0.8450	0.8375	0.8700	0.825625
128	0.6475	0.8200	0.8425	0.7275	0.8000	0.8275	0.7950	0.7975	0.782187

Table 3. Classification results of MAE for dataset 2 - Table 4 in [11].

Dimension	k_neighbors	svm_linear	svm_radial	decision_tree	random_forest	multi_layer	ada_boost	gaussian_nb	average
2	0.7150	0.5025	0.5800	0.7125	0.7000	0.5425	0.7000	0.7575	0.651250
4	0.8150	0.5975	0.7525	0.8150	0.8175	0.7425	0.8050	0.8050	0.768750
8	0.8300	0.5975	0.7025	0.7600	0.8100	0.7275	0.8025	0.7725	0.750312
16	0.8350	0.7475	0.8375	0.8125	0.8500	0.8225	0.8175	0.8625	0.823125
32	0.7775	0.7675	0.8300	0.7700	0.8150	0.8150	0.8150	0.8500	0.805000
64	0.7875	0.7600	0.8475	0.7900	0.8325	0.8200	0.8300	0.8650	0.816562
128	0.6500	0.8100	0.8275	0.7600	0.8100	0.8075	0.7775	0.8300	0.784062

Table 4. Classification results of MAAE for dataset 2 - Table 5 in [11].

Considering Dataset 2 and Tables 3, 4 we obtained similar results when compared with the results obtained by the original authors [1]. In general, the original paper obtained a maximum accuracy greater than those obtained by our reproduction implementation, but the average and unique values per dimension are close in most cases considering both functions AE-CDNN-MAE and AE-CDNN-MAAE. However, for Dataset 1 the accuracy values obtained in this paper are significantly lower than the ones obtained by the original paper considering both AE-CDNN-MAE and AE-CDNN-MAAE, as shown in Figure 6.

When analyzing the behavior of the different loss functions, we see that the MAAE function does not always obtain superior results than the MAE function. The same is observed in the original article, but their average accuracy obtained are higher than those obtained by our reproduction.

For each k-fold, we observed the accuracy values are following the class balance in the test. It's indicating the classification methods there are not better than random classifications that follow the classes' percentage. We observed below the result for the accuracy inspection, for the cross validation, for $m = 2$. Analyzing the accuracy obtained by classifiers in Tables 5 and 6 we observe the values obtained by AE-CDNN-MAAE and AE-CDNN-MAE are close, however the function AE-CDNN-MAAE obtained smoothly better results and with less variation, in general.

5-fold	k_neighbors	svm_linear	svm_radial	decision_tree	random_forest	multi_layer	ada_boost	gaussian_nb
1	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.4
2	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.4
3	0.4	0.6	0.6	0.6	0.6	0.6	0.6	0.4
4	0.4	0.6	0.6	0.6	0.6	0.6	0.6	0.4
5	0.4	0.6	0.6	0.6	0.6	0.6	0.6	0.4

Table 5. Classification results of MAE for dataset 1 - Table 6 in [11].

5-fold	k_neighbors	svm_linear	svm_radial	decision_tree	random_forest	multi_layer	ada_boost	gaussian_nb
1	0.71	0.6	0.6	0.67	0.64	0.6	0.69	0.43
2	0.71	0.6	0.6	0.70	0.68	0.6	0.74	0.41
3	0.81	0.6	0.6	0.70	0.77	0.6	0.75	0.43
4	0.75	0.6	0.6	0.72	0.65	0.6	0.73	0.44
5	0.71	0.6	0.6	0.68	0.73	0.6	0.74	0.54

Table 6. Classification results of MAAE for dataset 1 - Table 7 in [11].

In the original paper we observe similar differences between the two functions, the results for AE-CDNN-MAAE are slightly better for most classifiers. But considering **gaussian_nb**, for example, the function AE-CDNN-MAAE obtained much better results compared to with AE-CDNN-MAE. Although the results in original paper also have few vari-

ations for the classifiers **svm_linear**, **svm_radial** and **multi_layer** we had no variations in these classifers for the autoencoderAE-CDNN-MAAE.
In the second dataset, in Tables below, when analyzing by fold we had worser results than those reported by the authors. However, the results is consistent with the hypothesis that during the process there was no feature learning. Also given the balance of this second dataset, we have that all methods do not present a better result than random chance.

5-fold	k_neighbors	svm_linear	svm_radial	decision_tree	random_forest	multi_layer	ada_boost	gaussian_nb
1	0.7750	0.5000	0.7250	0.8250	0.7625	0.3250	0.8125	0.850
2	0.7125	0.5000	0.5375	0.7375	0.6875	0.4500	0.7750	0.775
3	0.7125	0.5125	0.5250	0.7125	0.6250	0.3875	0.7250	0.750
4	0.7000	0.5000	0.5000	0.7250	0.6500	0.4750	0.7500	0.775
5	0.6875	0.5000	0.5000	0.7000	0.6375	0.6250	0.7000	0.700

Table 7. Classification results of MAE for dataset 2 - Table 8 in [11].

5-fold	k_neighbors	svm_linear	svm_radial	decision_tree	random_forest	multi_layer	ada_boost	gaussian_nb
1	0.7875	0.5125	0.7250	0.7750	0.7750	0.6250	0.7375	0.8750
2	0.6875	0.5000	0.5375	0.7250	0.7125	0.4875	0.6875	0.7375
3	0.7375	0.5000	0.5750	0.7000	0.7000	0.4125	0.7125	0.7125
4	0.6875	0.5000	0.5250	0.7000	0.6375	0.6375	0.6625	0.7500
5	0.6750	0.5000	0.5375	0.6625	0.6750	0.5500	0.7000	0.7125

Table 8. Accuracy in Classification, with loss MAAE, on each fold cross-validation, for Dataset 2.

When analyzing the reduced values by class, specifically with $m = 4$, we note that3 of the 4 attributes are 0, in the best scenario, indicating that there was no learning in Auto Encoder to distinguish the behavior by class. This bad representation of latent space occurs regardless of the loss function.
When we analyze the behavior of the loss functions at the epoch, in Figure 8, in the first dataset, we note that our loss values are much higher than those obtained by the authors.
These differences also occur in the establishment in the baseline methods, indicating that there is some cut in the training set that was not included in this modeling, given the lack of information in the article. In Figure 9 we observe similar average accuracy between AE-CDNN-MAE, AE-CDNN-MAAE, PCA and SRP for both datasets.
The same is observed in the original article, as well as a similar behavior, but the average accuracy obtained for Dataset 1 are significantly higher than those obtained by our reproduction, as shown in the Table 9:

Dimension	k_neighbors	svm_linear	svm_radial	decision_tree	random_forest	multi_layer	ada_boost	gaussian_nb	average
16	0.790	0.6	0.600	0.746	0.786	0.586	0.760	0.584	0.6815
32	0.802	0.6	0.602	0.722	0.828	0.712	0.792	0.638	0.7120

Table 9. Accuracy in Classification, in dataset 1 with $CV = 10$ - Table 9 in [11].

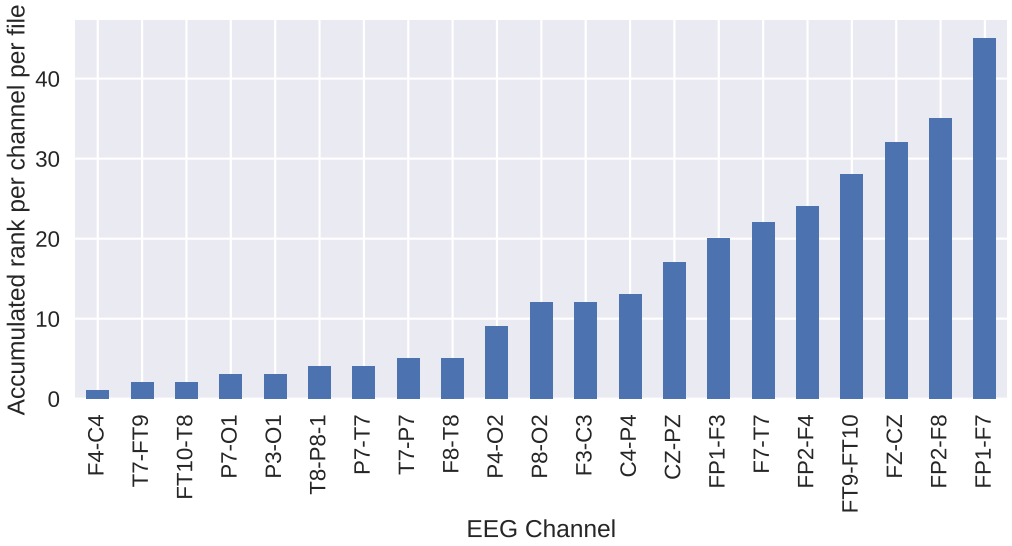
When we analyze the result assuming a 10-fold, we have an increase in the accuracy values for the first dataset, however, still below that reported by the original authors.

5 Conclusion

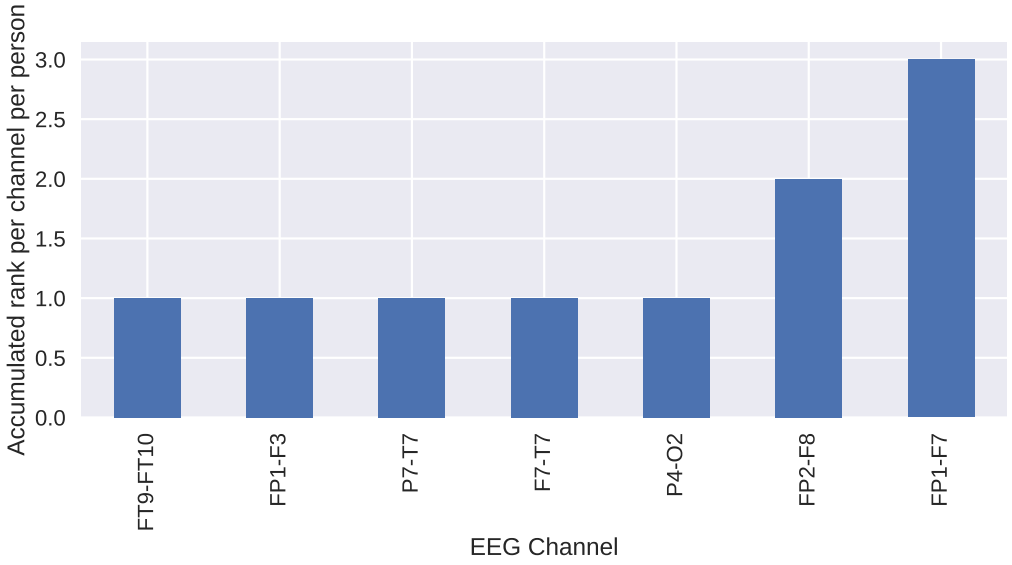
In this article, we re-implemented the approach proposed in [1] and propose the use of a different classifier. This classification approach, based on deep learning for detecting epileptic seizures using EGG had not been explored previously. We adopted a Auto-Encoder that allowed us to construct a smaller representation space.
The original authors left some gaps that made it impossible to fully reproduce their experiment. For example, the lack of information about the Section IV. C Comparison of Classification Application in the original paper, the sampling process of the first and second data sets and the number of times or batch size (see 2.1 to check the assumptions we made).
As contribution, the developed code can be easily ported to other tasks. Moreover, it could be used to evaluate other variants of the neural network architecture, techniques for classifying the signals, data augmentation, among other possibilities.

References

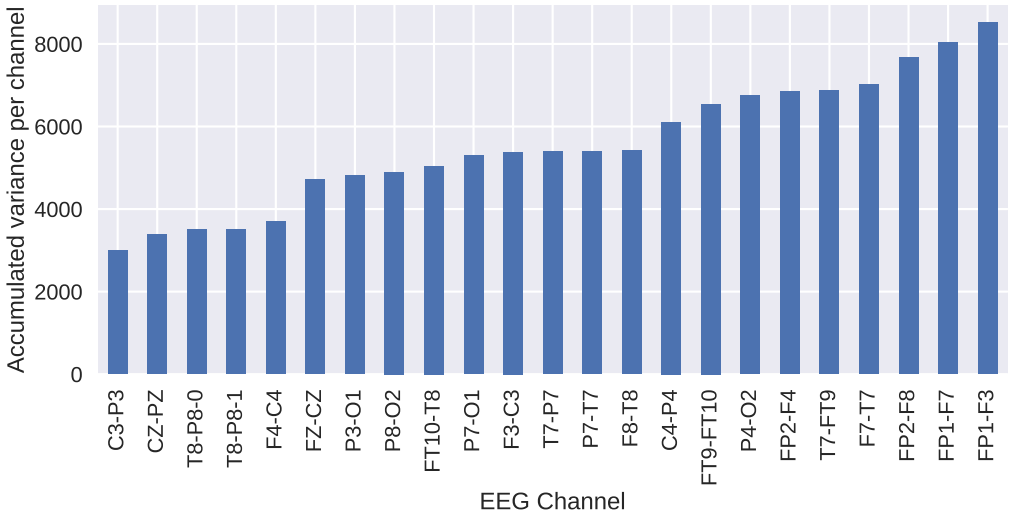
1. T. Wen and Z. Zhang. "Deep Convolution Neural Network and Autoencoders-based Unsupervised Feature Learning of EEG Signals." In: **IEEE Access** PP (May 2018), pp. 1–1.
2. F. Chollet et al. "Keras: The python deep learning library." In: **Astrophysics Source Code Library** (2018).
3. M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. "Tensorflow: A system for large-scale machine learning." In: (2016), pp. 265–283.
4. A. D. la Fuente and R. Aduviri. "[Re] Variational Sparse Coding." Python. In: **ReScience C** 5.2 (May 2019), #2.
5. A. Shoeb and J. Guttag. "Application of Machine Learning to Epileptic Seizure Detection." In: ICML'10 (2010), pp. 975–982.
6. A. Emami, N. Kunii, T. Matsuo, T. Shinozaki, K. Kawai, and H. Takahashi. "Autoencoding of long-term scalp electroencephalogram to detect epileptic seizure for diagnosis support system." In: **Computers in biology and medicine** (2019).
7. R. G. Andrzejak, K. Lehnertz, F. Mormann, C. Rieke, P. David, and C. Elger. "Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state." In: **Physical review. E, Statistical, nonlinear, and soft matter physics** 64 (Jan. 2002), p. 061907.
8. C. Kamath. "Analysis of EEG dynamics in epileptic patients and healthy subjects using Hilbert transform scatter plots." In: **Open Access Library Journal** 2.1 (2015), p. 1.
9. A. H. Shoeb. "Application of machine learning to epileptic seizure onset detection and treatment." PhD thesis. Massachusetts Institute of Technology, 2009.



(a) Accumulated variance per sample, considering a sample as each recording file.



(b) Accumulated variance per sample, considering a sample as being all the recordings of each person.



(c) Total accumulated variance in the first ten people of the dataset, as granular as possible.

Figure 1. Three scenarios modeled to calculate the variance.

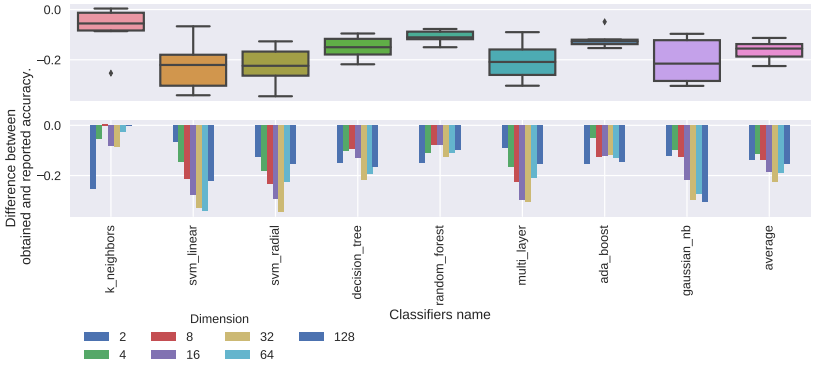


Figure 2. Classification Accuracy Results, for AE-CDNN-MAE as feature learning, in Dataset 1. Reproduced and Difference between the values contains in Table 2 in [1]. We use the same hyperparameters in the classifiers.

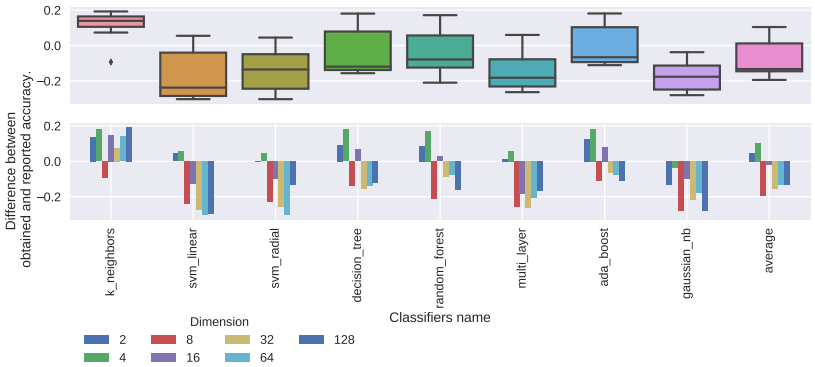


Figure 3. Classification Accuracy Results, for AE-CDNN-MAAE as feature learning, in Dataset 1. Reproduced and Difference between the values contains in Table 3 in [1]. We use the same hyperparameters in the classifiers.

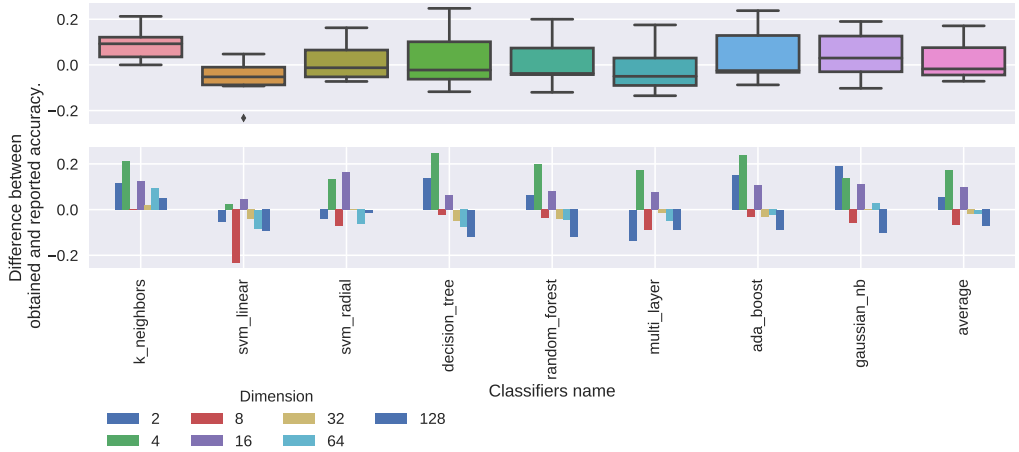


Figure 4. Classification Accuracy Results, for AE-CDNN-MAE as feature learning, in Dataset 2. Reproduced and Difference between the values contains in Table 4 in [1]. We use the same hyperparameters in the classifiers

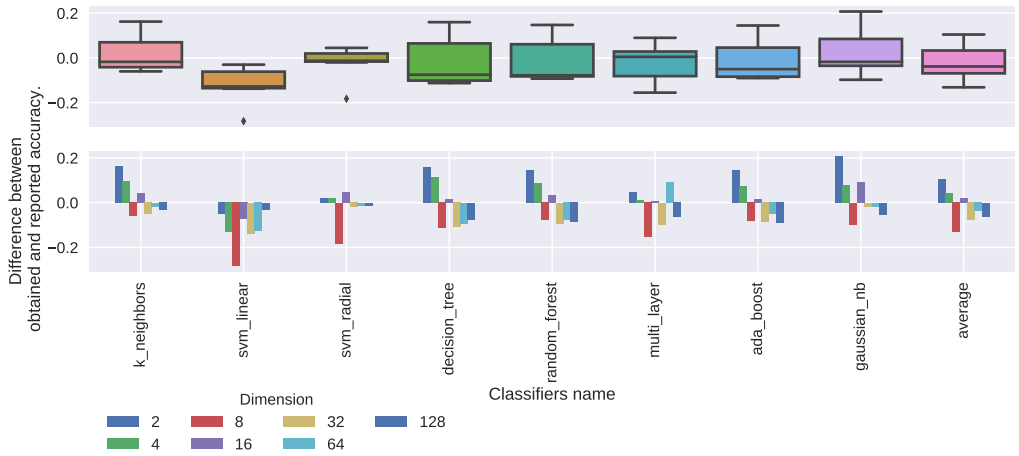


Figure 5. Classification Accuracy Results, for AE-CDNN-MAAE as feature learning, in Dataset 2. Reproduced and Difference between the values contains in Table 5 in [1]. We use the same hyper-parameters in the classifiers

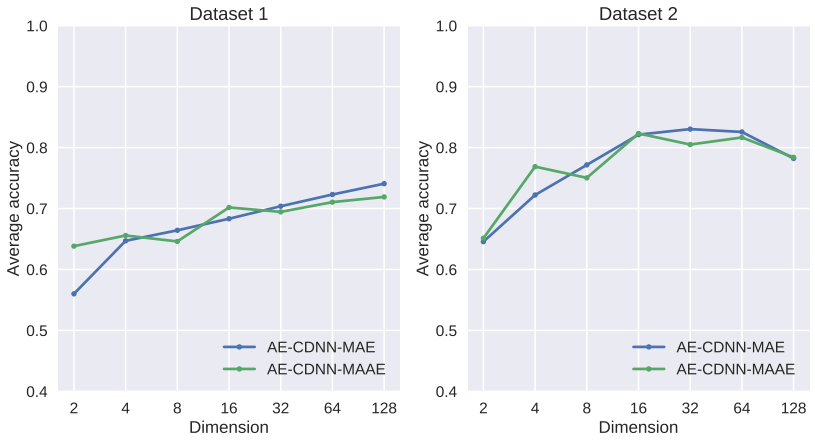


Figure 6. Average Accuracy Results of AE-CDNN-MAE and AE-CDNN-MAAE, with different dimension values in the two dataset. Reproduction of Figure 7 in [1].

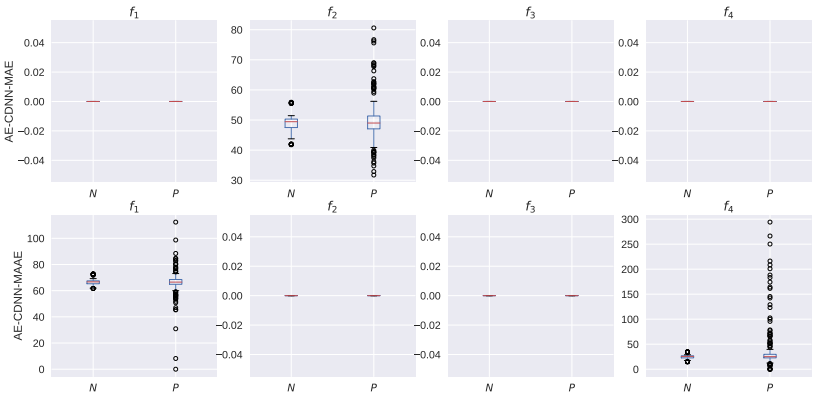


Figure 7. Feature Distribution of AE-CDNN-MAE and AE-CDNN-MAAE, with $m = 4$, in the first dataset. Reproduction of Figure 8 in [1].

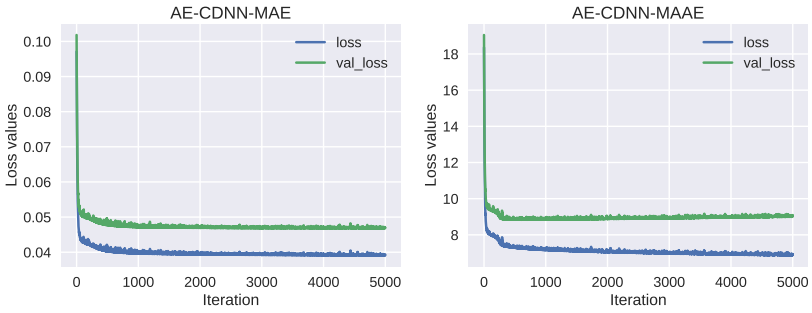


Figure 8. Change of loss function of AE-CDNN-MAE and AE-CDNN-MAAE, in the first dataset, with $m = 4$. Reproduction of Figure 9 in [1].

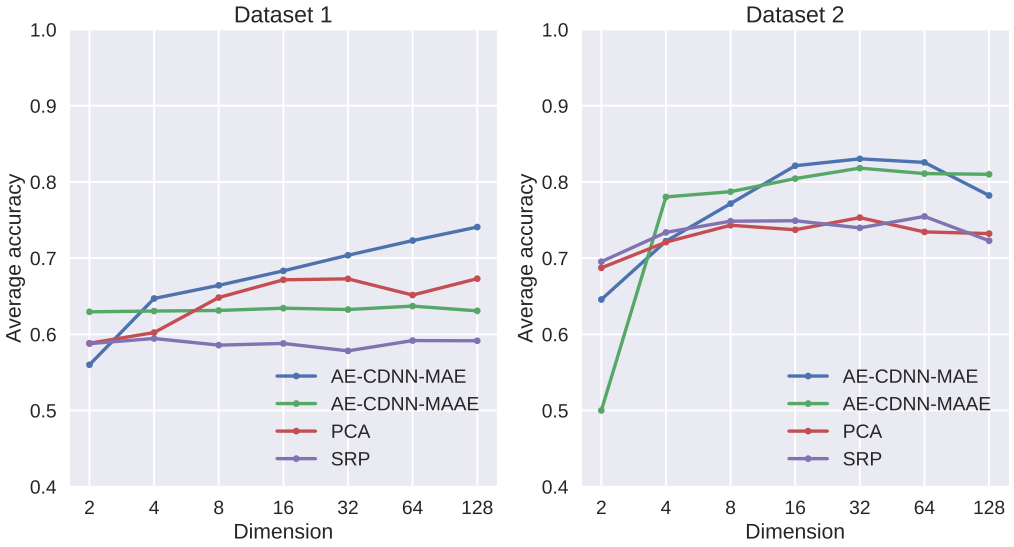


Figure 9. Comparison of accuracy for different loss functions (AE-CDNN-MAE, AE-CDNN-MAAE), and also with baseline (PCA, SRP). Reproduction of Figure 10 in [1].