AUTOMATIC FEATURE EXTRACTION FROM BIOSIGNALS USING CONVOLUTIONAL NEURAL MODELS

Jorge Torres[1], Julio David Buldain[2]

Instituto Universitario de Investigación
de Ingeniería de Aragón
Universidad Zaragoza

[1]Accenture
[2]Department of Electronic and Communication Engineering, University of Zaragoza

INTRODUCTION

Biosignals, such as electrocardiogram (ECG), electroencephalogram (EEG) or electromyogram (EMG), are signals collected from humans or animals that contain useful information about physiological phenomenons. For example, they can describe heart activity (ECG), brain activity (ECG) or muscle activation (EMG).

When working with biosignals, the first step is usually studying the signal to identify the most relevant points, which can be a time consuming task.

Our main goal is to achieve a convolutional neural model which can be used as a first step by classifiers, extracting the most relevant features of a biosignal and eliminating the need to study and select the relevant features to train the classifier.

To achieve this, we have trained a two dimensional convolutional model using ECG, EEG and EMG signals, so the model can learn to recognize features from biosignals.

METHODOLOGY

As each database has been created with signals recorded using different devices and sampling frequencies, our first step was to resample ECG, EEG and EMG signals to the same base frequency, 512 Hz. Then, we split these resampled signals into segments of 5 seconds to generate spectrograms from them, splitting them into training, validation and testing sets. We saved these spectrograms as images of size 640x480 pixels, which are fed as inputs to the convolutional model.

We built a convolutional neural network (CNN) to extract the relevant features from the input spectrogram. This model is formed by 4 convolutional layers, using 64 filters in each layer of size 3x3. After each convolutional layer, we added one 2x2 size MaxPooling layer to reduce the features from the convolutional layer by half, outputting 64 matrix with 30x40 elements in its last layer.
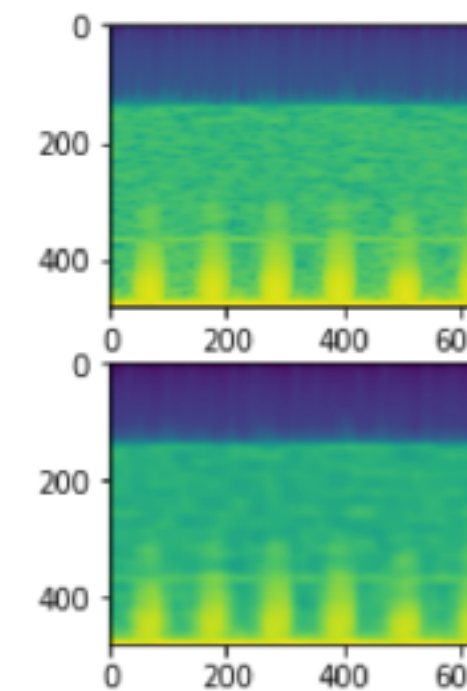
IMAGES

Fig 1. The upper image shows an ECG spectrogram send as input to the CNN model, whereas the bottom image shows the image reconstructed by the decoder
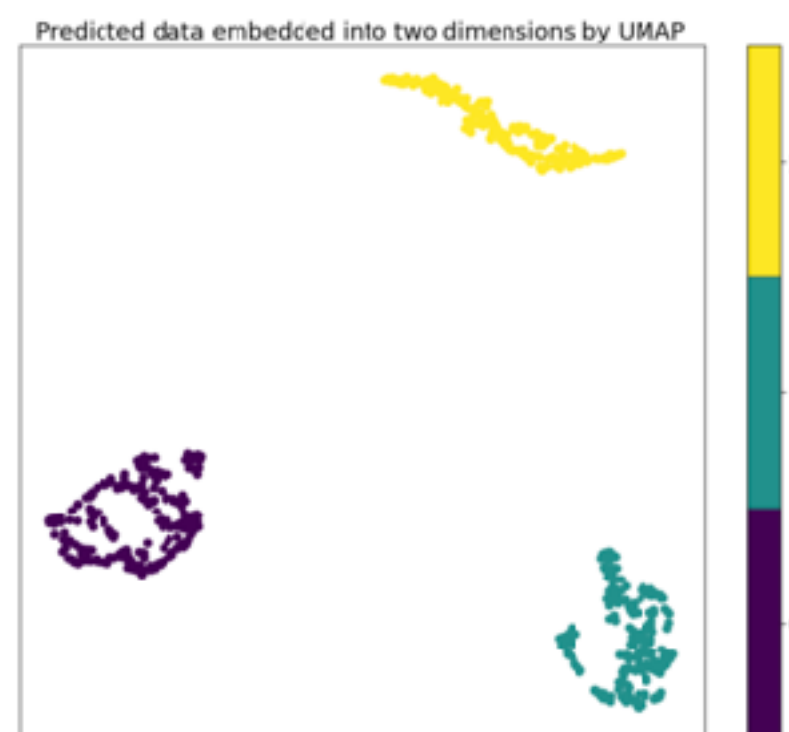

Fig 2. The image shows the two dimensional projections calculated by UMAP from features extracted by the encoder. Label 0 represents ECGS, 1 EEGs and 2 EMGs.

TRAINING

To train the model, we used an encoder-decoder structure, where we used the convolutional model as encoder and we appended a decoder to reconstruct the input image from the features extracted by the encoder. This decoder is built with transpose convolutional layers, using 64 3x3 filters per layer, and 2x2 upsampling layers to invert the operations from the encoder, generating a 640x480 image.

We trained the stacked model with 125.000 spectrograms generated from ECG, EEG and EMG signals. We chose Adam as the training optimizer and mean squared error to check training performance. Also, we used validation data sets to ensure that validation loss did not increase, preventing the model from overfitting.

Once the stacked model was trained, we used test samples to check if the output image was properly reconstructed. We fed several spectrograms to the model and, as the Fig. 1 shows, the model was capable of rebuilding an image similar to the original.

As another validation, we used UMAP to calculate a two dimensional representation from features extracted by the encoder, plotting them to verify if the features are spatially separated. First, we split the encoder structure from the stacked model and we fed it with 768 spectrograms (256 ECGs, 256 EEGs and 256 EMGs) to extract their features. We plotted their two dimensional representation generated by UMAP, as shown in Fig. 2, verifying that extracted features are grouped by each signal type.

CONCLUSIONS

We have trained a two dimensional convolutional model to automatically extract features from an input spectrogram, proving that these features present distinctive characteristics between each type of signal.

Our next steps are focused on building a one dimensional convolutional model and train it with biosignal samples to test both convolutional models, searching for the dimensional architecture that has a better performance extracting relevant features for a next stage of classifiers of different signals.