OMsignal Block 2 Report

Éric Girard
Université de Montréal
921969
e.girard@umontreal.ca

Yasmine Dumouchel
Université de Montréal
20113689
yasmine.dumouchel@umontreal.ca

Benjamin Rosa
Université de Montréal
1064709
benjamin.rosa@umontreal.ca

1 Introduction

The OMsignal project data set consists of electrocardiography (ECG) signals recorded through a set of sensors (recording module) embedded in clothes. The goal of this project is to predict, using those signal recordings, the identify the user as well as well as 3 metrics, namely the mean of the PR-Interval, the mean of the RT-Interval, and the standard deviation of the RR-Interval.

Using sensors is a very efficient way to collect a large amount of data efficiently. However, one challenge that arises with such a high amount of data is that it requires a lot of time to label them for supervised learning purposes. The data set consists of multiple ECG signals from 32 participants. More precisely, the labeled data consists of 15 windows per participant and the unlabeled data consists of 657,233 windows. A window is a 30 seconds recording with 125Hz frequency. As we can see, the amount of unlabeled data greatly surpasses the amount of labeled ones. For this reason, our goal is to leverage those unlabeled data and develop an unsupervised pre-training approach.

Several attempts of classification on ECG can be found in the literature. Shweta H. Jambukia (2015) use an Artificial Neural Network to classify ECG signal to diagnoses heart diseases. Another paper Essam H. Houssein (2017) introduce an ECG heartbeat classification based on Water Wave Optimization and Support Vector Machine. The classification is made to differentiate patient with life-threatening and non-life-threatening arrhythmia's. More recently, a paper Zhaohan Xiong and Zhao (2018) used convolutional recurrent neural network also in an attempt to detect cardiac arrhythmias.

Even if the previous publications used ECG, it was not for the same purpose as our project. Lugovaya (2014) was the only literature reference we found regarding human identification based on ECG. This paper was also referenced by Nicolas Laliberté (2019) for their use of PCA.

To approach the task at hand, we first focused on data pre-processing. By looking at different papers stated above, we identified that the usual pre-processing steps done on this kind of data often consisted of noise removal, normalization, data augmentation, Fast Fourier Transform, and log spectrogram. We ended up using two transformations which are detailed in section 2.1 of this report.

We then focused on implementing an system to make the best use of the unlabeled data. Several models such as an Autoencoder, a Mean teacher, a Virtual adversarial training, and the use of pseudo labels, were available options. Ultimately, we chose the unsupervised pre-training approach, which consisted of implemented an Autoencoder trained with the unlabeled data. The resulting trained Encoder section of the Autoencoder, would then be used to get an latent representation of our labeled data.

Finally, we had to choose the global architecture of our model and answer several questions such as the use of a multitasks neural net (such as refered in this paper : Inlong Ji (2018)) or on the contrary, the use of several neural nets, one for each tasks. In the end, we chose to implement a multitasks neural net.

In the last part of this report, we will discuss our methodology and results for the model selection. Since our model have several hyperparameters we had to do some fine tuning to try and find the best values for them, using a validation set. Our results are discussed in the last section.

2 Methodology

2.1 Data pre-processing

We tried several pre-processing transformations such as shifting the signals, flipping the signals, and adding gaussian noise to the signals. However, none seemed to add value in the training. We thus, decided to not include these transformations in the end.

We used a normalization method to pre-process our data. this method comes from Nicolas Laliberté (2019) and as is stated in section 2 their paper: "*The digitized ECG samples come with noise and different levels of amplitude. In order to remove the low frequency noise, we apply a moving average with windows of size 2. At each step, we centralize the data by the mean computed in this windows. Similarly, we normalize the amplitude of the signal within a moving windows of size 4, dividing the element within by the standard deviation of the windows.*"

After this normalization step, we segmented our ECG signals through a process that divided the window into several paired heartbeat. The number of division may vary depending on the number of R peaks per windows. Each resulting sample however have a standardize size of 230. Further, every segment contains two R peaks, i.e. one RR-Interval. This transform was originally done to get better results on the regression task for the RR standard deviation estimation. This segmentation acts as a form of data augmentation as it increases the amount of data used. Our motive for segmenting the signals was to have less feature to pass to our prediction module. Further, by representing the signals as a pair of heartbeat, we thought it would facilitate the work of the predicting information related to the intervals.

Our original idea was to train an Autoencoder using the unlabeled dataset. The encoder allows us to get a latent representation of our signals and only keep the relevant features and then the decoder then recreates the original input. The encoder was then applied to the labeled data as a transform for the labeled data prior to entering the prediction module. Section 2.2.1 of this report outlines the architecture of this Autoencoder in more depth. In the end, unfortunately, this encoding transformation did not result in higher performance, and as such, was not kept in our finale model.

2.2 Model architectures

2.2.1 Leveraging unlabeled dataset

To leverage the use of unlabeled data we added, we trained an Autoencoder on the unlabeled data exclusively. We tried several configurations and randomly set several hyper parameters and kept the configuration having the best result, i.e. the best reconstruction loss. We chose to implement a Convolutional Autoencoder given that we wanted to capture features that were invariant to its position.

We tried three different architectures for the Convolutional Autoencoders, with respectively 1, 2, and 3 layers. The most performing architecture turned out to be one with the 3 layers Autoencoder. Table 1 and Table 2 bellow describes the architecture.

Layer	Kernel	Filters	Stride	Padding
ConV/BatchNorm/ReLU	13x13	16	1	6
Max-Pooling	2x2	-	-	-
ConV/BatchNorm/ReLU	9x9	32	1	4
Max-Pooling	2x2	-	-	-
ConV/BatchNorm/ReLU	5x5	64	1	2
Fully connected layer	-	-	-	-
Fully connected layer	-	-	-	-

Table 1: Final Convolutional Encoder Architecture

Layer	Kernel	Filters	Stride	Padding
Fully connected layer	-	-	-	-
Fully connected layer	-	-	-	-
Transpose ConV/ReLU	5x5	64	1	2
Max-Unpooling	2x2	-	-	-
Transpose ConV/ReLU	10x10	32	1	4
Max-Pooling	2x2	-	-	-
Transpose ConV/ReLU	13x13	16	1	6

Table 2: Final Convolutional Decoder Architecture

2.2.2 Prediction module

Now, for the core of our architecture, the prediciton module, we implemented a multi-tasks neural network. It's architecture is described in Table 2. As showm, the first five layers are shared by each tasks, i.e. regardless of the metric we are predicting, our samples all pass through these layers. The output layer however, is specific to the each individual prediction task.

Layer	Kernel	Filters	Stride	Padding
ConV/BatchNorm/ReLU/MaxPool	9x9	60	1	4
ConV/BatchNorm/ReLU	5x5	60	1	2
Linear/BatchNorm/ReLU/Dropout	-	60	-	-
Linear/BatchNorm/ReLU/Dropout	-	60	-	-
ID output layer : Linear/BatchNorm/ReLU	-	-	-	
ID output layer : Linear/BatchNorm/ReLU/Linear	-	-	-	
PR output layer : Linear/BatchNorm/ReLU	-	-	-	-
PR output layer : Linear/BatchNorm/ReLU/Linear	-	-	-	-
RT output layer : Linear/BatchNorm/ReLU/Linear	-	-	-	-
RT output layer : Linear/BatchNorm/ReLU	-	-	-	-
RR output layer : Linear/BatchNorm/ReLU	-	-	-	-
RR output layer : Linear/BatchNorm/ReLU/Linear	-	-	-	-

Table 3: Multi-tasks Prediction module Architecture

3 Results and discussion

To assess the capabilities of our Autoencoder, we visualized the data. To do so, we applied Principal Component Analysis (PCA) and reduced our signal to 2 and 3 dimensions on both the original data and the encoded data, as seen on Figure 1 and Figure 2.

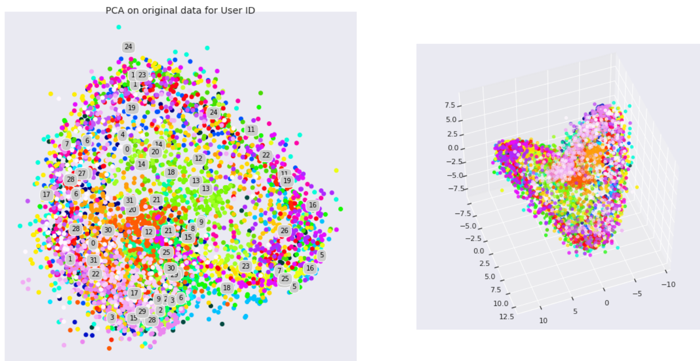

Figure 1: PCA on user ID of original data

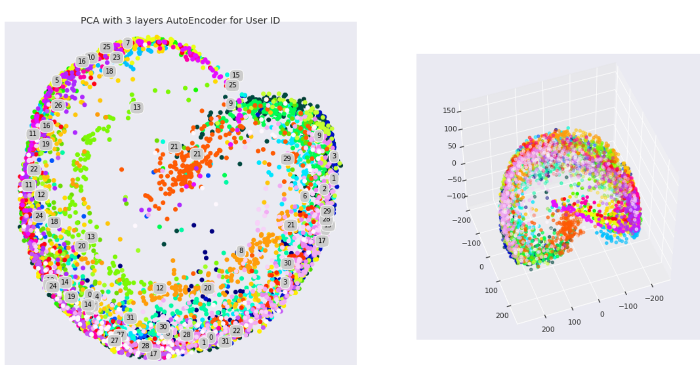

Figure 2: PCA on user ID of Encoded data

Figure 3: PCA on user ID of encoded data

Given that the figures turned out to be very crowded, we decided to plot a few pairs of user id, one against each other, to see if they formed distinct clusters. The resulting plots are shown in Figure 3. We ca observe that some user clearly have distinct clusters, such as user 3 and 5 or 6 and 2. Conversely, some, such as 9 and 2, seem to be a practically non-distinguishable.

Further, when using the encoder with the labeled data, the performance for all 4 tasks did not improve. We thus, decided to omit the encoding transform from our final model. Although this entailed not using the unlabeled data in our model, having segmented our data resulted in a much larger data set and resulted in performance which surpassed the baseline.

Finally, to further increase the prediction power of the individual tasks, we tested a few classical machine learning algorithms to the RR-Interval standard deviation predictions and see if it would beat our multi-task prediction module. By doing so, we were able to improve two metrics, namely the RR-Interval standard deviation and the User ID. For the former, we used segments of the average of two heartbeat and applied PCA, followed by a Support Vector Regression (SVR). And for the latter, we used the average of segments of one heartbeats applied PCA followed by Linear Discrimination Analysis (LDA). We believe the improved performance was due to the information that we could extract from average heartbeat for the ID user

The following table summarize the results we had for each tasks, compared to the baseline.

Table 4: OMSignal results for each tasks

Model	PR_Mean	RT_Mean	RR_Std	User_ID
baseline	65.99%	76.33%	18.25%	27.74%
Autoencoder and Multitask	42.03%	73.23%	34.47%	23.75%
Multitask	74.24%	84.93 %	37.56%	64.37%
Multitask and PCA-LDA/SVR	74.24%	84.93 %	70.75%	78.85%

4 Conclusion

We had to implement different models to better perform on each individual tasks. There is still room for improvement. We also wanted to try a pseudo-labels approach but we ran out of time. We wanted to used it along with our transformations : normalization and segmentation from Nicolas Laliberté (2019) as well as the encoding . First step would have been to encode the unlabeled data then do a prediction on it using the prediction module. If the prediction is made with enough confidence (above a predefined threshold), the data would be added to the labeled dataset. Artificially increasing the dataset that way would have maybe led to better result. In our future research, we would like to implement a convolutional denoising autoencoder such as the one implemented in Keiichi Ochiai (2018). In this publication the authors used it for the classification of Arrhythmia but we think it can be used for our identification task as well.

References

Essam H. Houssein, M. K. (2017). Ecg signals classification: a review. *Intelligent Engineering Informatics*, 5:376–396.

Inlong Ji, Xuhui Chen, L. C. P. L. (2018). A deep multi-task learning approach for ecg data analysis. *IEE*, 10.1109/BHI.2018.8333385:124–127.

Keiichi Ochiai, Shu Takahashi, Y. F. (2018). Arrhythmia detection from 2-lead ecg using convolutional denoising autoencoders. *KDD*, pages 1–7.

Lugovaya, T. S. (2014). Biometric human identification based on ecg. *Master thesis LETI*.

Nicolas Laliberté, Ishaan Kamur, H. L. (2019). Omsignal block 1. *report*, pages 1–7.

Shweta H. Jambukia, Vipul K. Dabhi, H. B. P. (2015). Classification of ecg signals using machine learning techniques. *IEE*, 10.1109/ICACEA.2015.7164783.

Zhaohan Xiong, Martyn P Nash, E. C. V. V. F. M. K. S. and Zhao, J. (2018). Ecg signals classification for the detection of cardiac arrhythmias using a convolutional recurrent neural network. *Physiological Measurement*, 39.