ECG Heartbeat Classification: A Deep Transferable Representation

Paul-Louis Delacour, Mathieu Chevalley, Tristan Meynier, Daniel Garellick

March 2020

1 Samples visualization and clustering

We used the TSNE algorithm to reduce the number of dimensions from 187 to 2 in both data sets. Due to the significant difference in class sizes of the MIT dataset we sampled 600 signals per class at random which is roughly the size of the smallest class. The 2D plots of the MIT and PTB datasets are shown in Figure 1a and 1b accordingly. For the PTB data set the separation between the normal and abnormal cases is really noticeable. For the MIT data set, some classes are clearly separated from each other but there are some classes with conflicting points. This is indicative of the results. When comparing F1 scores achieved for each dataset it is seen that it is generally 7-8% higher.

2 Solve both datasets with RNNs

To solve both datasets using RNNs, we tried different recurrent gate types, and we found that a SimpleRNN works well for both datasets. Both models have two RNN layers and one last Dense layer. We used more hidden units for MIT than for PTB (a good engineering rule seemed to be "total number of parameters = numbers of datapoints").

For MIT, we got an accuracy of 95.77% and for PTB an accuracy of 87.36%, an AUROC of 83.49% and an AUPRC of 89.05%. These accuracy's are lower than the baselines. This is probably because a Conv based network is able to use information from both past and future timesteps, whereas an RNN based network only uses past timesteps. This intuition is also confirmed by the higher score we got with Bidirectional RNNs (see later).

3 Compare with other models

3.1 Bidirectional RNN

We implemented bidirectional RNNs to use the entire signal (past and future data) to make predictions. We got higher scores than with RNN which strengthens our intuition that using information about the whole signal generally improves prediction. Even with a simple model of two bidirectional layers followed by a dense layer gives an accuracy of 98.51% and 90.18% for the MIT and PTB data sets respectively. We also get an AUROC of 89.85% and AUPRC of 93.38% for PTB. The results are not far from the baselines.

3.2 CNN with Residual Blocks

We replicated the algorithm implemented by the authors of [1]. In the article residual blocks were used in a five-layer deep CNN. The code from kaggle as produced by N. Mine [2] was used as a starting point to replicate the articles algorithm and results. The input and output layers must have the same dimensions. We set the padding to "Causal" and "Same" for the aforementioned reason. Causal padding implies that the output at "t" does depends on the input at "t+1". The model is similar in most ways to the CNN baseline model with the exception of the residual blocks. Indeed this method seems to give the best results. For the MIT dataset - "causal" padding: F1 of 93.06%, Accuracy of 98.81%, "same" padding: F1 of 92.85%, Accuracy of 98.78%. For the PTB dataset - "causal" padding: F1 of 99.01%, Accuracy of 99.21%, AUROC of 98.94%, AUPRC of 99.15%, "same" padding: F1 of 99.4%, Accuracy of 99.52%, AUROC of 99.36% and AUPRC of 99.54%.

3.3 Suport Vector Machine (SVM)

We tried two different approaches to the application of SVMs to the dataset. We first applied an SVM on the whole dataset directly which is equivalent to applying an SVM to a 187-dimension feature matrix. Each signal is interpreted by the SVM as a point in 187-dimensional space. Indeed this approach seems to outperform the handpicked features described below, but took eight times longer. The results for the MIT dataset are - F1 of 95.17% and Accuracy of 94.4%.The results for the PTB dataset are - F1 of 95.77%, Accuracy of 94.02%, AUROC of 94.26% and AUPRC of 96.3%.

The second method, was to perform a feature extraction on the given dataset. We identified features that were relevant to the classes. Features that are of medical interest and features that were found to differentiate well the classes were extracted. Usually when medical professionals are called to assess an ECG they have multiple consecutive heartbeats such that they can calculate features such as the variance in heartbeat rate (RR-peak interval) as an indication of arrhythmia. Our dataset was formed of single hearbeats and hence limited our choice of features. The features included the positions and height of the peaks, the total length of the signal, the gradient at various relevant locations as well as the proportion of the signal above and below a given threshold. These were tuned to better differentiate between the signals. The results for the MIT dataset are - F1 of 83.03% and Accuracy of 77.9%.The results for the PTB dataset are - F1 of 86.15%, Accuracy of 81.76%, AUROC of 84.32% and AUPRC of 90.41%.

The soft-margin SVM selected implements an RBF kernel whose hyperparameters were calibrated using a grid-search. To account for the imbalance in the classes we used the in-built *sklearn* option for balanced weighing of the classes. We furthermore fitted an SVM as the classifier of the last convolutional layer of the baseline code. This proved difficult to tune for both good precision and recall in the given time-frame.

4 Ensemble of models

We used ensemble methods to increase the performance of our predictions. Two methods were implemented to combine model predictions:

- the mode function to take the most commonly predicted class over the combined models
- the softmax function to take the class with highest overall probability over the combined models

Method	Mode	Softmax
F1 score	98.61%	98.64%
Accuracy	98.67%	98.69%

Table 1: Results for the ensemble model {RNN + RNN bidirectional + CNN with residual blocks} for the MIT data set

Method	Mode	Softmax
F1 score	96.57%	98.06%
Accuracy	95.07%	97.20%
AUROC	94.32%	96.51%
AUPRC	96.14%	97.56%

Table 2: Results for the ensemble model {RNN + RNN bidirectional + CNN with residual blocks} for the PTB data set

Table 1 and 2 shows that the results are in any cases better than the one obtained using a single model. We get approximately the same scores as for the baseline.

5 Transfer learning

As suggested in the tutorial, there are different types of transfer learning.

- Transfer 1 : Transfer learning with RNNs, frozen base model.
- Transfer 2 : Transfer learning with RNNs, retraining whole model.
- Transfer 3 : Transfer learning with RNNs, first frozen base model, then re-training whole model.

Score	Transfer 1	Transfer 2	Transfer 3
F1 score	69.21%	87.42%	89.1%
Accuracy	78.36%	89.9%	91.24%

Table 3: Comparison of scores for different types of transfer learning

6 Summary of Results

Method	Accuracy	F1 score
Baseline	98.31%	91.07%
Resid Blocks	98.78%	92.85%
Resid Blocks Causal	98.81%	93.06%
RNN	95.77%	77.53%
RNN bidirectional	98.51%	91.54%
SVM (feat extract)	77.90%	83.03%
SVM (original data)	94.40%	95.17%

Table 4: Comparison of all methods for MIT dataset

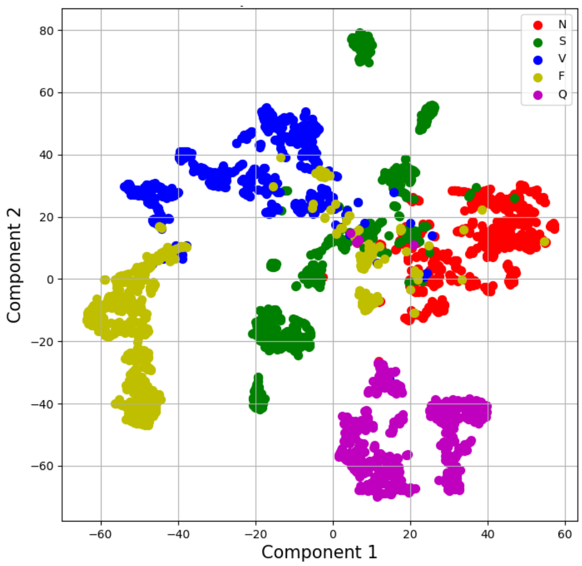
Method	Accuracy	F1 score	AUROC	AUPRC
Baseline	99.35%	99.55%	99.09%	99.34%
Resid Blocks	99.51%	99.40%	99.36%	99.54%
Resid Blocks Causal	99.20%	99.01%	98.84%	99.15%
RNN	87.36%	91.33%	83.49%	89.05%
RNN bidirectional	90.18%	93.01%	89.85%	83.38%
SVM (feat extract)	81.76%	86.15%	84.33%	90.41%
SVM (original data)	94.02%	95.77%	94.26%	96.30%

Table 5: Comparison of all methods for PTB dataset

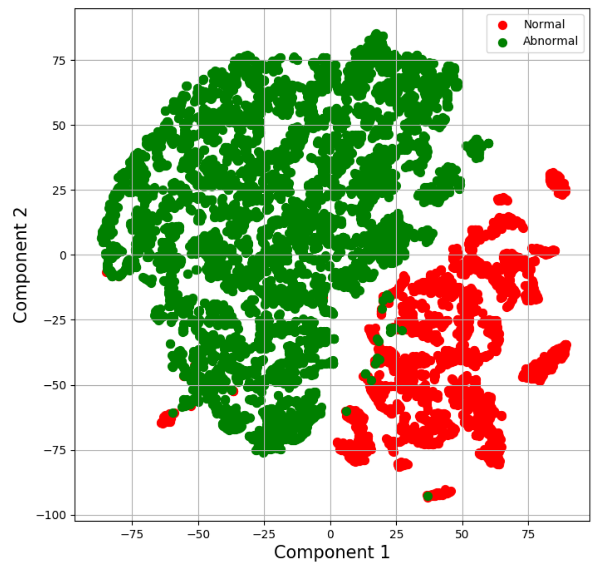
7 Reproducibility

Any results from the report can be obtained by running the *main* jupyter notebook. Score metrics are displayed as output. All the code used is available outside the *main* folder. A `conda` environment can also be found in the *main* folder.

TSNE Plots



(a) TSNE applied on MIT data set



(b) TSNE applied on PTB data set

Figure 1: Sample Visualisation and Clustering

References

[1] M. Kachuee, S. Fazeli and M. Sarrafzadeh, "ECG Heartbeat Classification: A Deep Transferable Representation," *2018 IEEE International Conference on Healthcare Informatics (ICHI)*, New York, NY, 2018, pp. 443-444.

[2] N. Mine, "Model from ArXiv 1805.00794", *coni57 - https://www.kaggle.com/coni57/model-from-arxiv-1805-00794*.