Multi-Stream Deep Neural Network For 12-Lead ECG Classification

Martin Baumgartner, Alphons Eggerth, Andreas Ziegl, Dieter Hayn, Günter Schreier

AIT Austrian Institute of Technology, Graz, Austria

Abstract

Advances in artificial intelligence and computer science have allowed for powerful assistive tools in a wide range of fields. Decision support systems could help health professionals to provide patients with quick and cost-efficient diagnostic analysis. The 2020 CinC Challenge challenges participants to develop such a tool for 12-lead ECG recordings.

In this paper, an approach for a multi-stream neural network is presented. Two parallel models were trained with different input data to combine the two relevant paradigms in modern machine learning. A simple multilayer perceptron and a deep convolutional neural network were concatenated for the final classification. Since the data originated from different sources, an ensemble of models was trained.

Due to technical difficulties, we (easyG) submitted a trimmed version and achieved a test score of -0.290, which ranked as the 39th entry. Validation score was 0.403. Although these results were mixed, offline 5-fold cross validation showed the potency of the full version.

Our results indicate that deep learning methods could in fact benefit from the addition of features derived via classical signal processing.

1. Introduction

Researchers have successfully – and impressively – demonstrated that machine learning can be used to fulfil complex tasks in healthcare. To give recent examples, groups have classified skin cancer [1], pneumonia [2] or antibiotic resistance in bacterial infections [3] on expert levels of accuracy.

1.1. 2020 CinC Challenge

The 2020 Computing in Cardiology (CinC) Challenge asked participants to correctly diagnose 12-lead electrocardiographic (ECG) recordings according to SNOMED coding [4]. The organizers provided 43.101 samples with 27 relevant labels from different sources as publicly available training material (see Table 1).

Table 1. Training datasets and sizes

Source	No. of samples
CPSC	6.877
CPSC-Extra	3.453
INCART	74
PTB	516
PTB-XL	21.837
Georgia	10.344

1.2. Multi-Stream neural networks

In machine learning two major paradigms emerged over time, that differ in when and how information is being distilled from raw data. The first option is to manually extract features with signal processing algorithms and provide them to models which then recognize patterns in them. This requires extensive knowledge about the data and the research field around it but simplifies the later modelling and increases transparency. The second approach is to "outsource" the information extraction to the model itself by developing computationally more powerfully – yet also more expensive – models. As this is mostly done with neural networks with many layers, this is referred to as "deep learning" and learning patterns are not as transparent.

The following chapters explain one way to combine both schools of thought that so often divide the research

2. Methods

We developed two neural networks that work in parallel before being concatenated by a final sigmoid-activated classification layer as depicted in Figure 1. The model was then trained as one combined model.

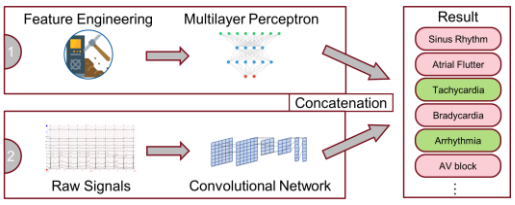


Figure 1. Multi-stream network as a schematic overview

 ISSN: 2325-887X DOI: 10.22489/CinC.2020.148

2.1. Feature stream

The feature engineering was based on previous work of the authors' research group [5]. The following Table 2 shows categories of features and the quantity of features in each category. Features were derived for every lead separately.

Table 2. Feature categories and quantities as in [5]

Category	No. of classes
Averaged beat	60
Beat	46
Atrial	39
Rhythm	205
Signal	6
QRS	16
Meta	14

The resulting feature matrices were processed by a multilayer perceptron (MLP) with 2 layers, of which the first layer had 1024 nodes (with dropout = 0.08) and the second 128 nodes (dropout = 0.1). Each layer was activated by a leaky rectifying linear unit and concluded with a batch normalizing layer.

2.2. Deep convolutional stream

The deep convolutional neural network (DNN) was based on the 2018 China Physiological Signal Challenge (CSPC) winners' model [6]. This model consists of 5 1D-convolutional blocks (see Table 3). Layers were activated by leaky linear rectifying.

Table 3. Description of one convolutional block (* the 5th block's last layer used a kernel size of 48)

Layer	No. of filters	Kernel size	Padding
1D-conv.	12	3	*same*
1D-conv.	12	3	*same*
1D-conv.	12	24*	*same*
Dropout = 0.2	-	-	-

The final block was followed by a bidirectional gated recurrent unit with 20 nodes (dropout = 0.2) and an attention layer, followed by a batch normalization layer.

The signals were filtered with a Butterworth bandpass filter (0.1 – 30 Hz) of 2nd order before being processed by the convolutional neural network.

2.3. Concatenation layer

A single fully connected layer with sigmoid activation concluded the multi-stream model so that the outcome would represent a probability for each label. The number of nodes was adapted to the number of used classes. After classification, the resulting vector was mapped to the full class vector to achieve a standardized outcome length for the final classification result.

2.4. Model ensemble and class reduction

As the datasets were from different sources, individual models were built. The rationale behind this, was the fact, that the records were labelled by different experts and with different pools of classes, likely resulting in diverging diagnosing criteria. Individual models could potentially compensate for these varying ground truths. Furthermore, this allowed for the reduction of classes overall. As stated above, 27 classes in total were considered for challenge, but since not every class was prevalent in each dataset, a model only has to learn those, which are in this specific dataset. A generic model would have to consider all classes, dramatically increasing the complexity of this task.

An unknown number of samples in the hidden test set are from additional, unidentified sources. In such cases, the entire ensemble was asked, and a majority vote would determine the final classification outcome.

To further reduce the dimensionality of this problem, extremely rare classes (under 1% of total labels of a dataset) were also not considered.

2.5. Training and evaluation routine

All dataset specific models were trained in a 5-fold cross validation process in 3 variations: features only, signals only and multi-stream (combined) model. In all cases the same optimizer (*Adam*), learning rate (*0.001*, $\beta_1 = 0.9$) and loss function (*binary crossentropy*) and model architectures were used for comparable results. The 4 larger datasets (CPSC, CPSC-Extra, PTB-XL and Georgia) were trained for 200 epochs, while the 2 smaller sets (PTB and St. Petersburg INCART) were trained for 500 epochs to compensate for their smaller training size.

During training for every fold, the challenge metric was calculated after each epoch with the evaluation code provided by the organizers [4]. Highest achieved metrics were also recorded. The x-axes in the Results section's figures were normalized to [% of epochs] to compare the processes of different lengths.

3. Results

3.1. Official CinC Challenge Score

Unfortunately, we (team: easyG) were not able submit our full approach for the challenge leaderboard because of

technical burdens associated with our approach. To score a valid entry for the challenge, we submitted a single generic DNN model, without feature engineering. Our final highest achieved metric is thus low at 0.403 (validation) and -0.290 (test), which ranked as 39th out of 41. We encourage readers to look beyond this limitation and examine the results achieved in offline validation presented below (Table 4).

3.2. Highest achieved metrics

The results presented in this chapter are achieved with the 6 publicly available offline datasets and constitute the highest achieved challenge metric (described in [4]) in any of the 5 folds during offline cross validation.

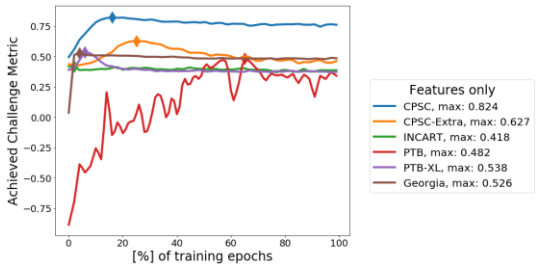
Table 4. Highest challenge metrics for the 3 model variations for each dataset achieved in training in any fold

Dataset	Features	Signals	Combined
CPSC	0.837	0.840	0.844
CPSC-Extra	0.687	0.426	0.659
INCART	0.678	0.513	0.560
PTB	1.000	-0.048	1.000
PTB-XL	0.549	0.518	0.552
Georgia	0.535	0.582	0.549
Weighted Average	**0.608**	**0.571**	**0.612**

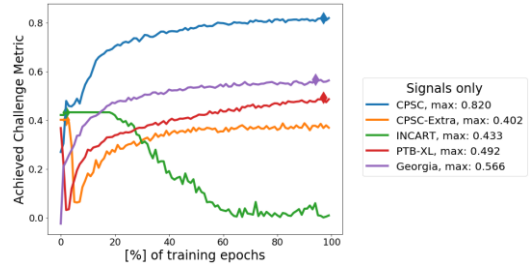
3.3. Optimization course

Figures 2a-2c illustrate the course of achieved metric during the offline training process for the 3 variations. The curves are averaged over all 5 cross validation folds.

a)



b)



c)

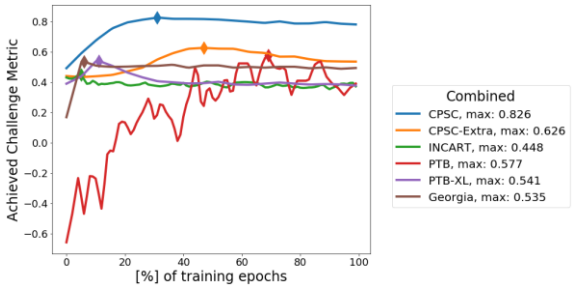


Figure 2. Optimization course for the 3 variation of models: a) features only, b) signals only, c) combined (PTB data removed in b) since it produced no scores > 0). Highest mean scores are additionally noted in the legend. Diamond-shaped markers indicate when highest metric was achieved during the optimization process.

4. Discussion

Although an entry with a complete version of our idea into the challenge's official leaderboard was ultimately not possible due technical difficulties, there are still interesting aspects to be learned from our approach. Combining two different machine learning paradigms leads to novel challenges, some of which are presented here. The weighted averages in Table 4 can be interpreted as hypothetical leaderboard scores with the limitation that samples from unknown, additional sources could not be included in this offline validation.

4.1. Optimization disparity

As stated earlier, the two streams operate with inputs of varying information density. Features are information dense, as the extraction of knowledge is done manually prior to the model training, while the DNN receives unprocessed, raw data and thus is required to extract patterns itself. This leads to a disparity in optimization peaks between the two streams, where the feature stream's MLP is optimized significantly earlier than the DNN as seen in differences in peak timings between Figure 2a and Figure 2b. Thus, determining training duration was a challenging task and was further complicated in this experiment as the feature stream showed a tendency to overfit. Ultimately, this led to the feature stream dominating the entire multi-stream network, as the DNN was not able to match the MLP's performance early on and never effectively learned. This was likely further pronounced by the asymmetrical concatenation, as the MLP had 128 nodes compared to the DNN's 40 nodes. Figure 3 shows an example of optimization disparity where the "*signals only*" model optimized noticeably later than the "*features only*". Similar behavior was found in other datasets too, although, not as pronounced as in the CPSC set.

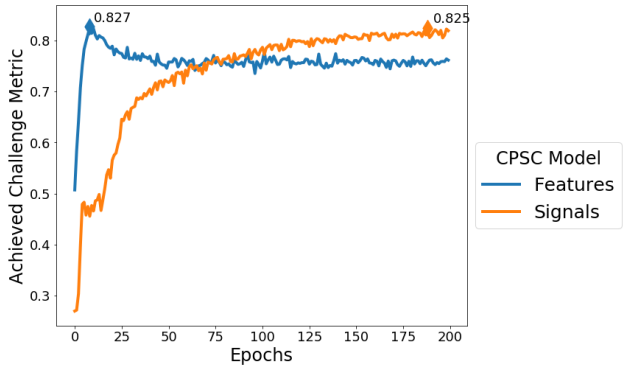


Figure 3. Optimization disparity shown with the example of CPSC model with an averaged optimization course for the "*features only*" (blue) and "*signals only*" (orange) model variant. Highest achieved scores for each model is annotated by diamond-shaped markers.

A possible solution to this issue could be to freeze the MLP's weights after a certain amount of training epochs and thus give the DNN chance to contribute more. Another option would be training both streams separately, transferring the weights into a combined but untrained model afterwards and only training the final concatenation layer as a form of transfer learning. This approach seems reasonable, yet the combined effect and synergistic effects could be reduced without joint training. Naturally, a symmetrical concatenation could also establish a better balance between streams.

Overall, the DNNs' streams could have benefitted from more training epochs, as Figure 2b shows a tendency of optimization peaks at later stages of the training process.

4.2. Dataset differences

As the Results section showed, the same approach performed vastly different on all datasets. While the model for the CPSC dataset performed well, the two large datasets (PTB-XL, Georgia Database) were significantly less well classified. This might seem counter-intuitive but can be explained by two facts. Firstly, the CPSC set is the 3rd largest dataset but only had 6 different classes, which constitutes a balanced mix between amount of training samples and degrees of freedom in classification. While the 2 larger datasets benefit from a high number of training samples, the also had a higher number of significant classes (PTB-XL: 19, Georgia: 20). This likely increased the problem's complexity to the extent, that the model was overwhelmed. Secondly, the model architecture had been optimized during the unofficial phase where only the CPSC dataset had been available. As could be expected, the small datasets either produced distinct training instability (PTB) or virtually no training effect at all (INCART). Although the underlying type of data is the same in all datasets, quantitative and qualitative variations in classes have enough impact that individual models can potentially produce better results. This reinforces this paper's approach, but also highlights the importance of model architecture adaptation.

4.3. Conclusion on synergistic effects

The highest weighted averaged was achieved by the multi-stream model with a score of 0.612 outperforming both single stream networks. This confirms our original idea, that a combined model could perform better than its parts. However, a critical look at Table 4 also reveals that not all datasets benefitted from the combined approach. In the CPSC and PTB-XL sets, the combined model performed better, while this effect was not seen in other datasets. In general, the performance of the feature stream was more stable than the DNN stream. Furthermore, in all but one case (Georgia) the addition of features to the DNN lead to an increase in score. This suggests, that deep learning methods could likely benefit from the addition of features in general. Additionally, the MLP's stream also constituted only an estimated 11.8% of the combined models' computational effort, meaning extending DNNs with a feature stream would also be a cost-effective improvement. These results encourage us to pursuit this method further, improve on identified weaknesses and test its effectiveness in future research.

References

[1] Esteva A et al., "Dermatologist-level classification of skin cancer with deep neural networks", *Nature*, vol. 542, no. 7639, pp. 115–118, 2017.

[2] Ng A et al., "CheXNet: radiologist-level pneumonia detection on chest x-rays with deep learning", *arXiv.org*, 2017.

[3] Khaledi A et al., "Predicting antimicrobial resistance in Pseudomonas aeruginosa with machine learning-enabled molecular diagnostics", *EMBO Molecular Medicine*, vol. 12, no. 3, 2020.

[4] Perez Alday EA et al., "Classification of 12-lead ECGs: the PhysioNet/Computing in Cardiology Challenge 2020", *Physiological Measurement*, 2020.

[5] Kropf M et al., "Cardiac anomaly detection based on time and frequency domain features using tree-based classifiers", *Physiological Measurement*, vol. 39, no. 11, 2018.

[6] Chen TM et al., "Detection and classification of cardiac arrhythmias by a challenge-best deep learning neural network model", *bioRxiv*, 2019.

Address for correspondence:

Martin Baumgartner
Austrian Institute of Technology, Giefinggasse 4, 1210, Vienna
martin.baumgartner@ait.ac.at