Machine Learning for Health Care

Project 4

Mesut Ceylan, Sven Gregorio, Moritz Vandenhirtz

June 2021

1 Introduction

We are given two separate data sets that have a varying number of observations and signal classes. There are two crucial key points observed regarding the given dataset. First, both data sets show class imbalance, thus requiring well-tailored models to handle them. Second, the main property of heartbeat signals is the sequential structure in which the ordering within observations plays a key role in distinguishing signal classes. With that in mind, we provide various approaches to tackle the binary and multi-class classification problems of heartbeat signals.

2 Task 1: Vanilla Models and Baseline Comparison

In this task of the project, we experiment with the given baseline model, baseline Multi Layer Perceptron, Vanilla Convolutional Neural Network and Recurrent Neural Network to solve the classification task for both datasets and report the performances of each model. Table 2 depicts the model performances and comparison for Task 1.

2.1 Vanilla MLP

As the received baseline is already optimized and well designed for the task, we decided to create our own baseline with a leaner architecture, to explicitly measure the improvements of our other models. Thus, the Vanilla MLP only consists of a fully connected layer with 256 neurons and ReLu activation.

2.2 Vanilla CNN

To gain insights about the data and design architectures that function well, we start with a simple convolutional neural network(CNN) that follows a similar structure of the received baseline. We determined the main hyperparameters using Bayesian Optimization. As regularization we use early stopping and dropout.

2.3 Vanilla RNN

Following the Vanilla CNN, we also experimented with a simple recurrent neural network (RNN) consisting of 128 Simple RNN units accompanied with Dense and Dropout layers. We utilized masking such that the model does not take non-informative padding into account that was used to make the data equally long. Due to computational complexity, we decided that an extensive hyperparameter search is neither feasible to conduct nor the main focus. Yet, the Vanilla RNN performance is worth to report in order to be able to conduct a comparison and analysis.

2.4 Task 1 Results

According to the model performances depicted in Table 2, it is evident that the Baseline model is well-designed and tuned for the tasks at hand, especially for the MIT-BIH dataset. However, the Vanilla models depicted show promising performances without any complex architecture, specifically the Vanilla CNN outperforms the baseline model on the PTB dataset. Therefore, we allocate our focus to design well-tailored and high performing CNN and RNN based approaches to tackle other tasks of the project.

3 Task 2: Additional Models

In this task of the project, we introduced differing models w.r.t to the network architecture and units, hyperparameters, loss functions and methodologies.

3.1 Fourier Transform Based CNN Model

As we realized that the data is padded to have equal length sequences we tried to transform the signal in a way which hides the gaps from the model. Thus we apply a Fourier transform on the signals after replacing the padding with the value 0.5, as the original zero-padding introduced stronger artifacts. We then train the baseline model on the transformed input.

3.2 Bidirectional LSTM

LSTM units are specially designed to process sequential data structures such as signals. They are highly capable of maintaining a long-term sub-structure dependencies within sequences and their special gate structures allows them to obtain control over the information flow throughout the each unit. Most importantly, by introducing bidirectionality to the LSTM units, we enable LSTM units to process each sequence from both direction, beginning to end and vice versa, to capture time dependencies of the signals. Since heartbeat signals are sequential and time-dependent, we believe these properties of LSTMs are a crucial aspects to utilize for signal type classifications.

Thus, we designed a network consisting of 128 Bidirectional LSTM units along with Dense and Dropout layers, of degree 0.1. We trained this network with the Sparse Categorical Cross Entropy Loss function and with the Adam optimizer. We again use a Masking to stop the model from using the padded values.

3.3 CNN with Residual Connection

Apart from Bidirectional LSTM units, we were also inspired by a deep residual network architecture, the powerful ResNet. It is proven that deeper network architectures perform very well in combining low-mid-high level features to the network and avoid the vanishing/exploding gradient problem when residual connections among convolutional layers are established. Additionally, these residual connections enable the feature transfer from preceding layers to succeeding layers, combining extracted feature representations throughout the architecture. We benefit from this deep neural network with residual connections by distilling and transferring the extracted features from different layers of the architecture which allows us to tackle the class imbalance and capture dependencies of the sequential data.

3.4 Dilated CNN + Bidirectional LSTM

To improve performance and robustness of Bidirectional LSTM model, we introduced hybrid architecture consisting of Convolutional and Bidirectional LSTM layers. The main idea of this model is to first extract and distill the crucial features from the sequences by utilizing 1D Dilated Convolutions and then process the distilled and compressed features by a Bidirectional LSTM for classification. Instead of simply feeding input sequences to the Bidirectional LSTM units, we benefit from dilated convolutions to capture longer dependencies and useful features. We use a larger kernel field of view, a Maxpooling layer to reduce feature dimension throughout 4 convolutional blocks and Dropout layers to gain robustness. Additionally, we benefit from the Sparse Categorical Focal Loss function, a specially designed loss function that enforces model to learn hard negatives, in our case under represented observations, to combat the class imbalance and tackle the overconfidence problem of the Cross Entropy Loss.

3.5 Task 2 Results

Table 3 in the Appendix depicts the performances of the introduced models and baselines. The model performances depict that the deep CNN with Residual connection and the Dilated CNN with Bidirectional LSTM architecture outperforms the baseline model in both datasets.

4 Task 3: Ensemble of Models

For this task we combine the models we developed in ensembles and evaluate their performance. Ensemble methods have shown that the combination of multiple models reduces their variance and can lead to better results. We use three approaches for building ensembles: soft voting (predictions are averaged), hard voting (each model casts a vote) and logistic regression on the prediction of the models.

A priori we decided to use the logistic based approach with all models to compare it to the single models performances. Table 4 and 5 shows the performances of all Ensemble of models.

5 Task 4: Transfer Learning

Transfer Learning is a well-known approach of reusing models trained on a task for another task. Since the two dataset of the project have inputs with the same format and the first dataset has more data, we try to transfer some models from the first dataset to the second dataset.

For this task we experimented with two different transfer learning approaches and two different models. For the first approach we use the baseline model of the MIT-BIH dataset, replace its fully connected layers and only train those on the PTB dataset. For the second approach we use the 398-layer ResNet trained on the MIT-BIH dataset, replace its fully connected layer and retrain the whole model on the PTB dataset. Overall, we observed that Transfer Learning is very effective to further boost model performance.

6 Results

Overall, most of our models outperformed or are on par with the baseline model. We depict our model performances in Table 1 for the MIT-BIH and PTB test sets.

MIT-BIH Dataset	Accuracy
Baseline	0.9841
Vanilla MLP	0.9794
Vanilla CNN	0.9730
Vanilla RNN	0.9608
Fourier Model	0.9286
Bidirectional LSTM	0.9837
CNN with Residual Blocks	**0.9886**
Dilated CNN + Bidirectional LSTM	0.9856
Logistic Ensemble	**0.9894**

PTB Dataset	Accuracy	AUROC	AUPRC
Baseline	0.9794	0.9964	0.9981
Vanilla MLP	0.9732	0.9902	0.9948
Vanilla CNN	0.9811	0.9983	0.9994
Vanilla RNN	0.9138	0.9632	0.9828
Fourier Model	0.9859	0.9960	0.9970
Bidirectional LSTM	0.8691	0.9337	0.9727
CNN with Residual Blocks	0.9914	0.9986	0.9993
Dilated CNN + Bidirectional LSTM	0.9914	0.9963	0.9962
TL w/Retrained ResNet	**0.9958**	**0.9993**	**0.9997**
TL w/Frozen Baseline	0.9687	0.9957	0.9981
Logistic Ensemble	**0.9942**	**0.9997**	**0.9999**

Table 1: Evaluation of models trained on the MIT-BIH and PTB data set w.r.t to Accuracy, AUROC and AUPRC. The largest value of each column for the distinct models is **highlighted**. The Logistic Ensemble of all models is displayed separately.

7 Discussion

For the data preprocessing we experimented with many different approaches such as upsampling with SMOTEENN, Fourier transformation, truncated SVD but none of them showed promising results as they lead to a loss of information. Therefore, we decided to continue with the raw signal as input data and let the models generate adequate representations of the data. We explored many different classifiers such as Bidirectional LSTM, Dilated CNNs, CNN with Residual Connection, Transfer Learning and Ensembles. Through our big variety of approaches, we are able to hand in many different models that all perform very well. Combining them in an ensemble then shows the best results due to the variance reducing nature of the approach. We note that the Transfer Learning on the Retrained ResNet outperforms the Logistic Ensemble of all models in the Accuracy metric on the PTB dataset. However, we still believe that the Logistic Ensemble is the better model as the AUPRC is a more adequate metric in this case of heavy class imbalance.

8 Appendix

8.1 Task 1 Results

MIT-BIH Dataset	Accuracy
Baseline	**0.9841**
Vanilla MLP	0.9794
Vanilla CNN	0.9730
Vanilla RNN	0.9608

PTB Dataset	Accuracy	AUROC	AUPRC
Baseline	0.9794	0.9964	0.9981
Vanilla MLP	0.9732	0.9902	0.9948
Vanilla CNN	**0.9811**	**0.9983**	**0.9994**
Vanilla RNN	0.9138	0.9632	0.9828

Table 2: Evaluation of *vanilla* and *baseline* models trained on the MIT-BIH and PTB datasets w.r.t to stated performance metrics. The top performing model performances are **highlighted**.

8.2 Task 2 Results

MIT-BIH Dataset	Accuracy
Baseline	0.9841
Fourier Model	0.9286
Bidirectional LSTM	0.9837
CNN with Residual Connections	**0.9886**
Dilated CNN + Bidirectional LSTM	0.9856

PTB Dataset	Accuracy	AUROC	AUPRC
Baseline	0.9794	0.9964	0.9981
Fourier Model	0.9859	0.9960	0.9970
Bidirectional LSTM	0.8691	0.9337	0.9727
CNN with Residual Connections	**0.9914**	**0.9986**	**0.9993**
Dilated CNN + Bidirectional LSTM	0.9914	0.9963	0.9962

Table 3: Evaluation and comparison of *additional* models trained on the MIT-BIH and PTB datasets w.r.t to stated performance metrics. The top performing model performances are **highlighted**.

8.3 Task 3 Results

Top k models	k = 2	k = 3	k = 4	k = 5	k = 6	k = 7	k = 8
Soft voting	**0.9895**	0.9887	0.9879	0.9880	0.9872	0.9865	0.9858
Hard voting	0.9869	0.9879	0.9870	0.9873	0.9862	0.9863	0.9847
Logistic regression	0.9892	0.9893	0.9892	**0.9895**	0.9894	**0.9895**	0.9894

Table 4: Evaluation and comparison of ensemble methods on the MIT-BIH dataset w.r.t to accuracy. The top performing ensemble performances are **highlighted**.

Top k models	k = 2	k = 3	k = 4	k = 5	k = 6	k = 7	k = 8	k = 9	k = 10
Soft voting	0.99553	0.99622	0.99725	0.99656	0.99656	0.99691	**0.99759**	**0.99759**	0.99690
Hard voting	0.99209	0.99519	0.99725	0.99622	0.99725	0.99725	0.99690	0.99656	0.99553
Logistic regression	0.98351	0.72208	0.99347	0.99141	0.99038	0.99278	0.99381	0.99381	0.99416

Table 5: Evaluation and comparison of ensemble methods on the PTB dataset w.r.t to accuracy. The top performing ensemble performances are **highlighted**.