Report: Project 1

Machine Learning for Health Care

Remo Kellenberger, Michael Mazourik, Julian Neff

29.03.2022

1 Introduction

An electrocardiodiogram (ECG) is a simple test that can be used to check a patients heart's rhythm and electrical activity. An ECG is often used alongside other tests to help diagnose and monitor conditions affecting the heart by attaching a couple of sensors (electrodes) to the body. It can help detect a wide range of heart problems such as arrhythmias, coronary heart disease, heart attacks and cardiomyopathy (NHS, 2022).

For all our models one can find the hyperparameters and the settings we used inside our git repository (https://github.com/neffjulian/MLfHC).

2 Dataset

	Arrhythmia Dataset	PTB Diagnostic ECG Database
Number of samples:	109'446	14'552
Sampling Frequency:	125Hz	125Hz
Classes (Occurrence):	Normal (27.8%) Abnormal (72.2%)	Normal (82.8%) SEB (2.5%) VEB (6.6%) Fusion beat (0.7%) Unknown beat (7.3%)

3 Vanilla RNNs and CNNs

3.1 Solving with a CNN

Our vanilla CNN follows a traditional structure, we first have three filter layers followed by three fully connected layers. On each layer we use a ReLU activation function and in addition to that we use a small dropout. On the output layer we either used a sigmoid function followed by a binary cross entropy loss for the binary dataset or we used a cross entropy loss function with a softmax activation for the multiclass dataset.
Playing around with parameters we saw that the model starts to give reasonable predictions

when using at least two or three convolutional layers. Going deeper or wider with the convolutional layers did not lead to a significant improvement of the predictions.

3.2 Solving with a RNN

The vanilla RNN consists of a single LSTM unit with one layer and a fully connected layer to compute the class labels. The LSTM unit is initialized with a hidden state of 100 features. A basic ReLU activation function together with a dropout layer is applied before passing the data to the fully connected layer. The model was subject to a strange behavior shown in figure 3. The model was trained on a balanced dataset with equal number of samples per class.

3.3 Comparison to Baseline

Model	**Mitbih**	**Ptbdb**
Baseline	Acc: 0.979	Acc: 0.973 AUROC: 0.997 AUPRC: 0.998
Vanilla CNN	Acc: 0.975	Acc: 0.955 AUROC: 0.987 AUPRC: 0.995
Vanilla RNN	Acc: 0.727	Acc: nan AUROC: nan AUPRC: nan

4 Additional Models

4.1 CNN with Residual Blocks

For this model we built upon the structure of our existing CNN (3 convolutional layers followed by 3 dense layers) to get a direct comparison when adding residual blocks. From our testing we saw that adding one block to the second convolutional layer and adding one block to the second dense layer gave the best results.

Model	**Mitbih**	**Ptbdb**
CNN with Residual Blocks	Acc: 0.975	Acc: 0.955 AUROC: 0.987 AUPRC: 0.995

In addition to our two residual blocks we modified the blocks itself by either letting it run through a separat dense layer or by dividing the values of the block by some constant. It is usually a good idea to first normalize the values for the residual blocks (Naranjo-Alcazar et al., 2019) but with our approach we got (slightly) better results.

4.2 Bidirectional LSTM

The bidirectional LSTM model is based on the vanilla RNN with some additional features.

1. Convolutional layer. Two convolutional layers are applied before passing the data to the lstm units. The intuition behind this step is to reduce artifacts and increase information in the raw sequence. The two convolutional layers try to learn some features in the sequence before passing it to the lstm unit
2. Bidirectional LSTM unit. Bidirectional LSTM units allow for additional training by passing the data from left-to-right and right-to-left. Our model should benefit from this, because it allows to better learn long-term dependencies. Research has shown that Bidirectional LSTMs tend to have higher accuracy for prediction tasks with time series (Siami-Namini et al., 2019)

5 Ensemble Approaches

5.1 Adaboost Classifier (ABC)

Adaptive Boosting (Adaboost) is a multiplicative weight-update technique, or in simpler terms, a meta-learning algorithm which helps combine many weak classifiers Freund and

Schapire, 1997. The output of all the smaller and less accurate algorithms are combined using a weighted sum. In our instance, the hyper-parameters used for this algorithm are 50 bins and a learning rate of 1. Using the mitbih dataset, we achieved the following underperforming results:

ABC on Mitbih	**Training**	**Test**
Accuracy	0.863	0.865

5.2 Random Forest Classifier (RFC)

The random forest classifier is also a meta-estimator which combines many smaller decision trees to help prevent overfitting as well as improve accuracy. The idea of bagging is applied in a similar manner to combine noisy decision trees which tend to be noisy Hastie et al., 2001 In the case of a binary classification problem, the class is decided democratically by the majoriry of the trees. We had the following results with the Mitbih dataset:

RFC Classifier on Mitbih	**Training**	**Test**
Accuracy	0.999	0.974

6 Transfer Learning

For our transfer learning approach we were interested in seeing how a pretrained model which is specialised in image classification could help us. For this we chose ResNet-18 which is a CNN that is 18 layers deep. It is trained to classify up to 1000 different objects (He et al., 2015). Small modifications were made so we can pass the input in the form of an image.

Model	**Mitbih**	**Ptbdb**
ResNet-18 with small modifications	Acc: 0.919	Acc: 0.875 AUROC: 0.923 AUPRC: 0.969

7 Appendix

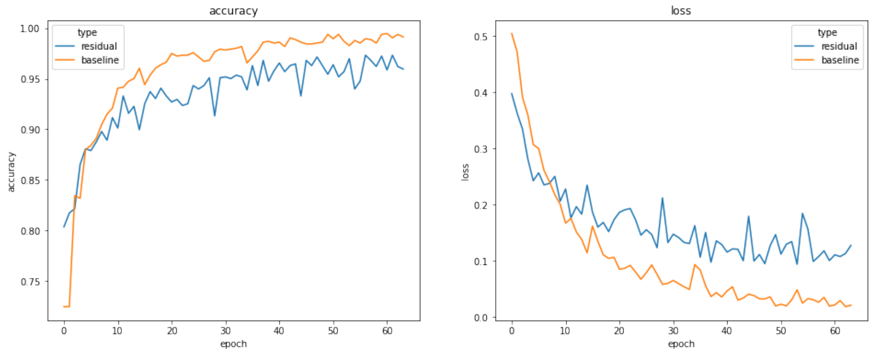

Figure 1: Validation loss and accuracy of Residual CNN on the PTB Diagnostic Dataset

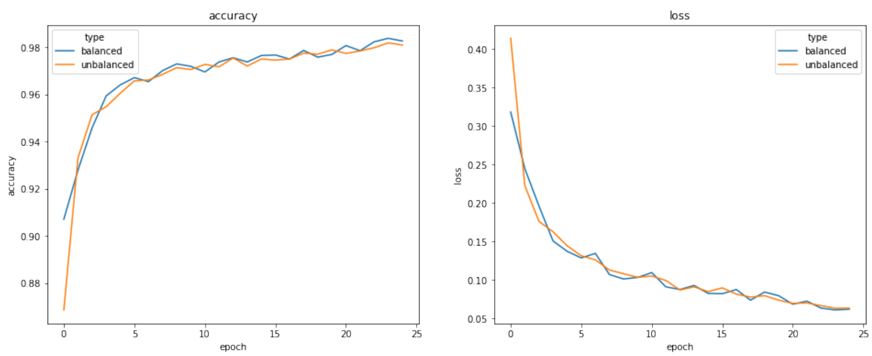

Figure 2: Comparison of balanced and unbalanced arrhythmia dataset with the baseline model

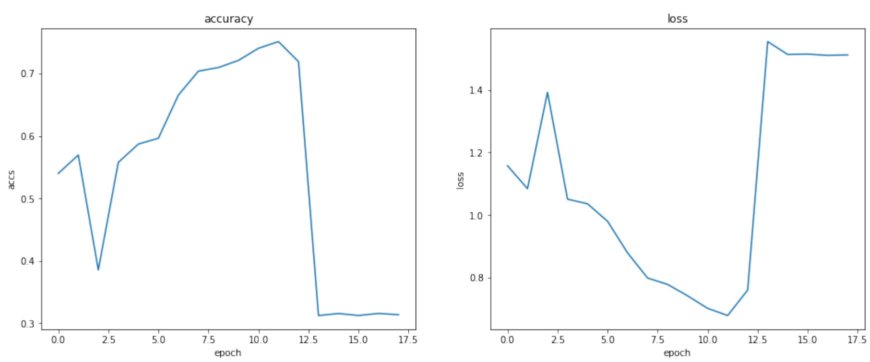

Figure 3: Accuracy and loss of vanilla RNN

References

Freund Y. and Schapire R. E. (1997). *A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting.* DOI: https://doi.org/10.1006/jcss.1997.1504.

Hastie T., Tibshirani R., and Friedman J. (2001). *The Elements of Statistical Learning.* Springer Series in Statistics. New York, NY, USA: Springer New York Inc.

He K., Zhang X., Ren S., and Sun J. (2015). *Deep Residual Learning for Image Recognition.*

Naranjo-Alcazar J., Perez-Castanos S., Martin-Morato I., Zuccarello P., and Cobos M. (2019). *On the performance of residual block design alternatives in convolutional neural networks for end-to-end audio classification.*

NHS (2022). URL: https://www.nhs.uk/conditions/electrocardiogram/.

Siami-Namini S., Tavakoli N., and Namin A. S. (2019). *The Performance of LSTM and BiLSTM in Forecasting Time Series.* DOI: 10.1109/BigData47090.2019.9005997.