ML4HC Report 1

Names: Flurin Hidber, Adriano Martinelli Nethz: hidberf, adrianom
Email: hidberf@student.ethz.ch, adrianom@student.ethz.ch IDs: 14-928-451, 15-917-404
Course: Machine Learning for Health Care, Prof. Gunnar Rätsch, Spring Semester 2019

Abstract

In this report we review our implementations of convolutional neural networks (CNNs), a network based on CNNs within residual stacked blocks and recurrent neural networks (RNNs) and compare their performances. We further use ensembles of these models to see if a combination of these techniques can yield an additional increase in the predictive capability. We extend our research by investigating the effects of transfer learning (TFL).

A Introduction

An electorcardiogram depicts the electrical activity of the heart and can be used as a proxy for the physiological state of the heart. Cheap, mobile and non-invasive ECG signal recording allows to collect large amounts of data in a short period of time and more importantly to deploy deep learning techniques to analyse the signals.

A.1 Data

Two famous heartbeat datasets were used in this project – the MIT-BIH Arrhythmia dataset and the PTB Diagnostic ECG database.

The MIT-BIH data categorises samples into 5 categories: Normal beat (class 0), Supraventricular premature beat (class 1), Premature ventricular contraction (class 2), Fusion of ventricular (class 3) and Unclassifiable beat (class 4) (see Figure 1).

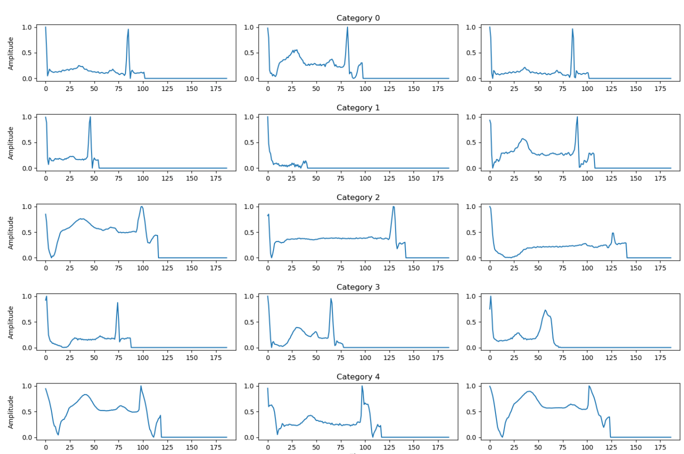

Figure 1: Visualisation of heartbeat samples of the MIT-BIH dataset

The PTB Diagnostic ECG dataset is composed of only two classes, the MI class and the normal class (see Figure 2).

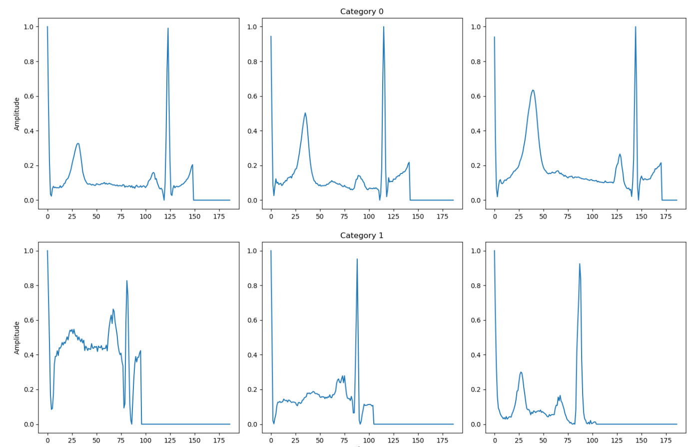

Figure 2: Visualisation of heartbeat samples of the PTB dataset

B Methods

There exist a number of analytical methods to analyze time-series data like ECG. Most involve two stages. First the time series data is transformed into a representation vector and the second, classifying that vector. Especially the first step can be difficult to engineer from human knowledge on the data topic (ECG) they are applied to. The development of deep learning methodologies in such data domains provide us with a new way to tackle learned feature engineering. Deep learning network architectures based on layers such as convolutional (CNNs) and recurrent (RNN) neural networks have proven to be able to create good representations to allow for predictive modelling in such domains.

B.1 Convolution Neural Networks (CNNs)

Our CNN network consists of two successive CNN layers followed by two linear layers with leaky rectified linear unit acitvation functions.

B.2 Convolution in Residuals (RCNN)

This network was implemented according to the model described by *M. Kachuee*, 2018. We used hyperparameter tuning to explore the possible best configuration, we provide on *github*.

B.3 Recurrent Neural Networks (RNN)

Our RNN approach uses LSTM layers (two successive ones with hidden size of 256), followed by a feedforward block consisting of two linear layers of size 1024 with leakyReLU activation before the output layer.

B.4 Model ensemble

For ensemble learning we used the CNN and the RCNN model architectures.

B.5 Transfer Learning

C Results

	CNN	RCNN	RNN
mitbih	0.982	0.957	0.987
ptbdb	0.278	0.298	

Table 1: Comparision of our three model types: CNN, RCNN and RNN.

	CNN + RCNN
mitbih	
ptbdb	

Table 2: Results of our ensemble method applied on the pretrained CNN and RCNN model.

	ptbdb	mitbih
freezing weights	0.279	0.959
/wo freezing	0.278	0.986

Table 3: Effect of transfer learning on the models capability on the other .

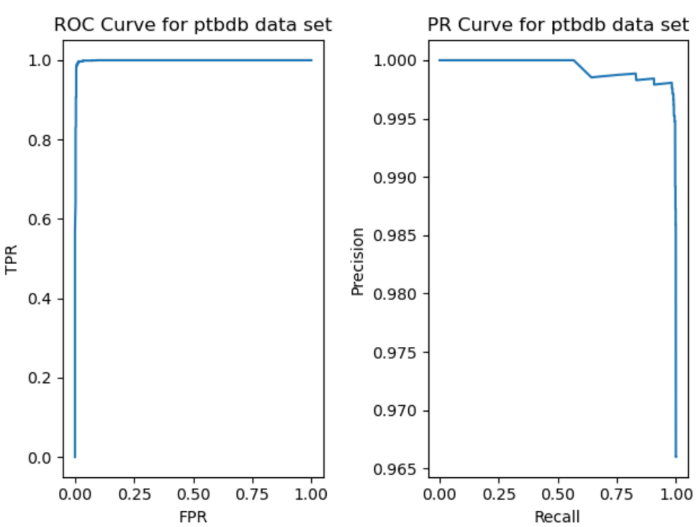

Figure 3: Results for the RCNN model on the PTB data.

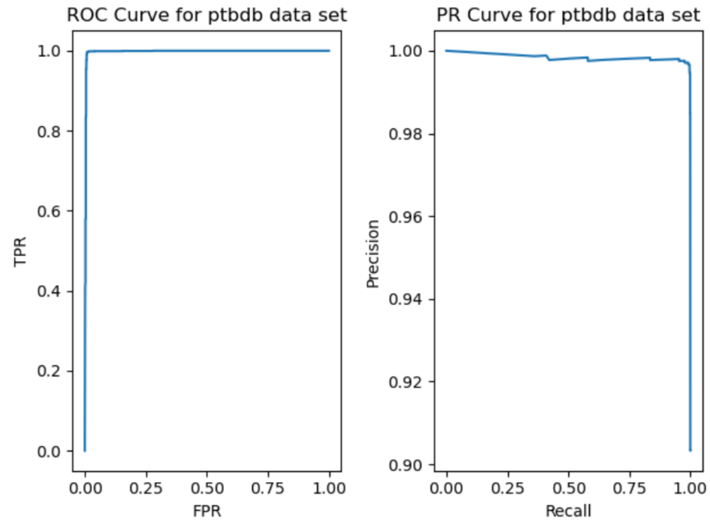

Figure 4: Results for the CNN model on the PTB data.

D Discussion

We were impressed by the results of our implementation of the recurrent neural network. We chose to go with a rather complex model which we hypothesize lead to over-fitting - poor results - on the smaller dataset (ptbdb-dataset). Ensemble methods showed success, suggesting that CNNs, RCNNs and RNNs are indeed complimentary in their predictions. Transfer Learning

D.1 Outlook

We wanted to combine the strength of both recurrent and convolutional neural networks, RNNs with their inert memory manage to capture long range dependencies very well, while CNN using temporal windows to apply kernels may be capable to denoise the signal. We propose a model using a convolutional layer in a first step to denoise the input signal, before processing the produced intermediary signal in a RNN network part, finally classifying after a feedforward network part with linear layers and non-linearities. (This was implemented, but - due to the limit on computational resources - we only managed to get preliminary results, which we did not report here.)

E References

[1] https://github.com/adrianomartinelli/machine-learning-for-health-care