Machine Learning for Health Care
ETH Zürich

Project 1

Omenetti Matteo, Ravi Srinivasan

March 27, 2022

Contents

1 Introduction

This report is part of the first project for the course "Machine Learning for Health Care" hosted at ETH Zürich. The scope of this project was to build different models for ECG classification for two different datasets PTB and MIT_BIH. All of the models described in this report can be found in "src/" and they have been trained (in a reasonable amount of time) using the GPUs/TPUs offered by Google COLAB-Pro. All of the images that summarize the achitecture of the different models have been omitted due to the length constraints of the report, but they can be found in "./models_architecture_images"

2 Convolutional Neural Network (CNN)

For the vanilla CNN, a smaller model compared to the one provided as a baseline was built. The reason behind this shallower architecture is that we were curious to understand whether the decreased training cost would also result in a statistically relevant decrease in performance. However, as it will be reported in the two subsections below this is not the case.

2.1 MIT_BIH

The code for this model can be found at "/src/MIT/CNN_mit.ipynb". The model on the Mit_Bih dataset achieves a test accuracy score of 0.981, taking around 6 minutes to train. This model was trained for 33 epochs with the option to reduce the learning rate on plateau, before early stopping kicked in. As image 1 and the reported accuracy show the performance of this shallower model are very close to the baseline model (accuracy 0.985) that instead takes more than 7 minutes to train on the same GPU.

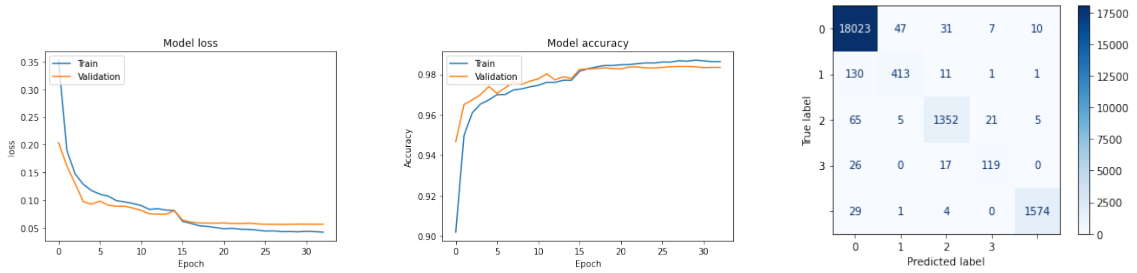


Figure 1: (left) train and validation loss of the vanilla CNN for the Mit_bih dataset.
(center) train and validation accuracy of the vanilla CNN for the Mit_bih dataset.
(right) confusion matrix of the vanilla CNN for the Mit_bih dataset

2.2 PTB

The code for this model can be found at "/src/PTB/CNN_ptb.ipynb". The model on the PTB dataset achieves a test accuracy score of 0.982, taking around 1 minute to train. As image 2 and the reported accuracy show, the performance of this shallower model is very close to the baseline model (accuracy 0.991). The AURPRC and AUROC curves can be found in image 3

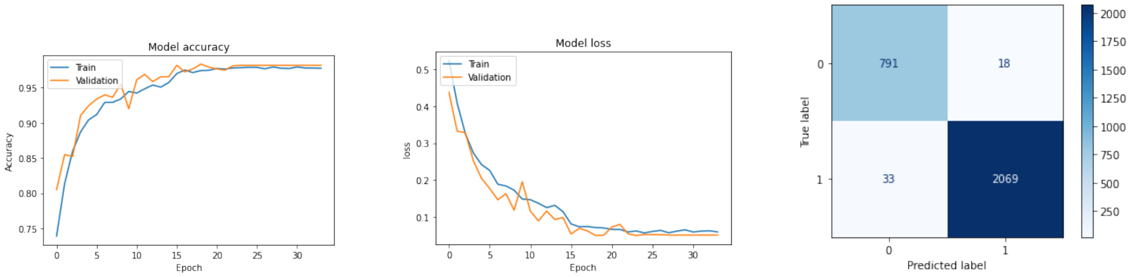


Figure 2: (left) train and validation loss of the vanilla CNN for the PTB dataset. (center) train and validation accuracy of the vanilla CNN for the PTB dataset. (right) confusion matrix of the vanilla CNN for the PTB dataset

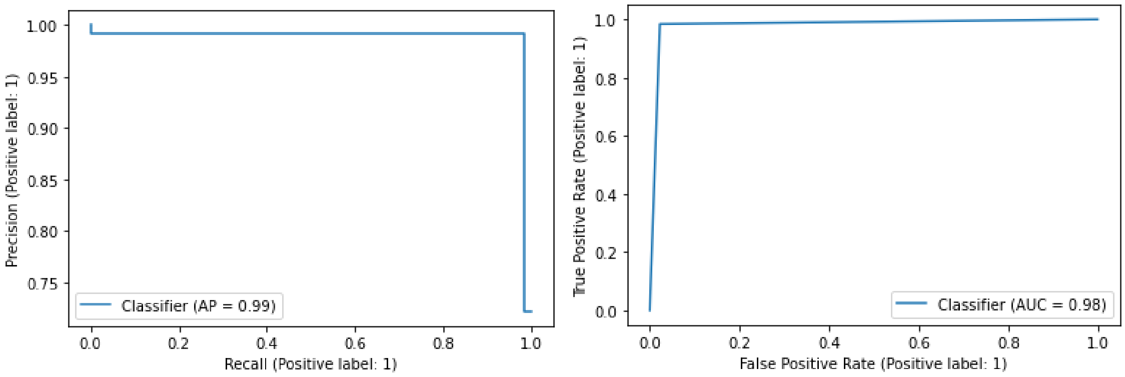


Figure 3: (left) AURPC curve of the vanilla CNN for the PTB dataset. (right) AUROC curve of the vanilla CNN for the PTB dataset

3 CNN with Residual Blocks

The code for this model can be found at "/src/MIT/ResNet_mit.ipynb". For this task, we designed a CNN architecture similar to ResNet (but shallower). This is the model that achieves the best performance. This model compared to the vanilla CNN adds residual blocks after each BatchNormalization layer.

3.1 MIT_BIH

This model on the Mit_Bih dataset, achives an accuracy of 0.988. The training curves and confusion matrix are summarized in image 4.

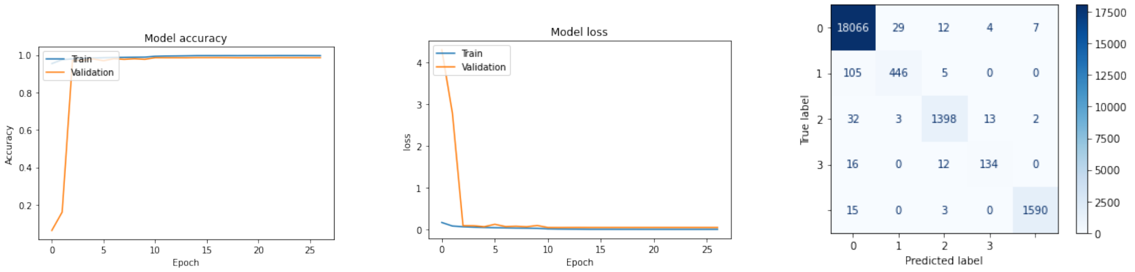


Figure 4: (left) train and validation loss of the CNN with residual blocks for the Mit_bih dataset.
(center) train and validation accuracy of the CNN with residual blocks for the Mit_bih dataset.
(right) confusion matrix of the CNN with residual blocks for the Mit_bih dataset

3.2 PTB

The code for this model can be found at "/src/PTB/ResNet_ptb.ipynb". This model on the PTB dataset, achives an accuracy of 0.986. The training AUROC, AUPRC curves and confusion matrix are summarized in images 5 and 6

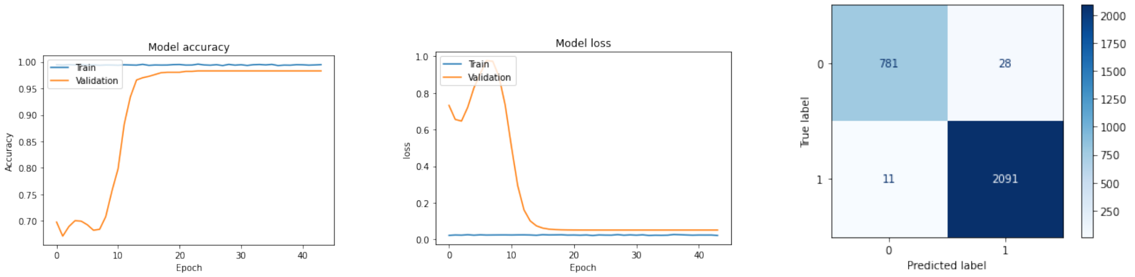


Figure 5: (left) train and validation loss of the CNN with residual blocks for the PTB dataset.
(center) train and validation accuracy of the CNN with residual blocks for the PTB dataset.
(right) confusion matrix of the CNN with residual blocks for the PTB dataset

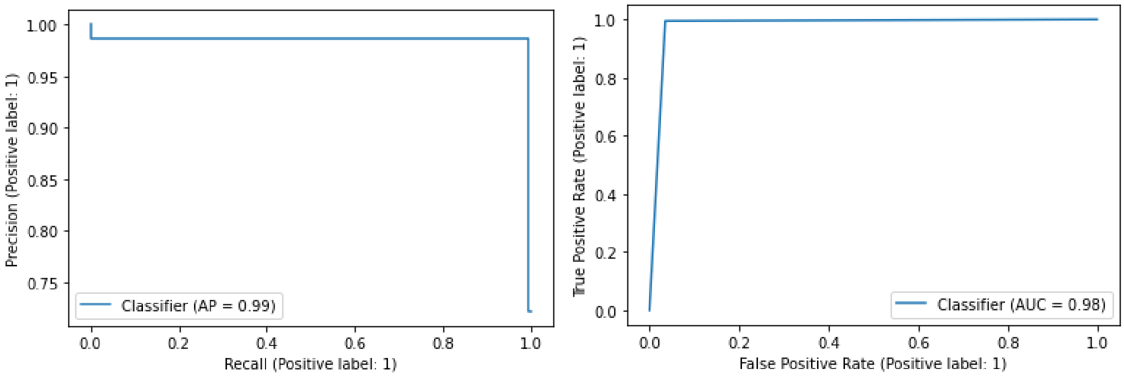


Figure 6: (left) AURPC curve of the CNN with residual blocks for the PTB dataset.
(right) AUROC curve of the CNN with residual blocks for the PTB dataset

4 Transformer

This model consists of three transformers encoder blocks, stacked on top of each other with a standard CNN at the end. This model is very heavy to train. Without ColabPro the training for the Mit_Bih dataset is around 16hours with the TPU. Using ColabPro, the amount of time required to train the model decreases drastically. Overall, its performance is not great if compared to the other much lighter models, therefore we would not recommend using this model in these particular tasks. Moreover, this model is also significantly slower than all of the other models at prediction time.

4.1 MIT_BIH

The code for this model can be found at "/src/MIT/Transformer_mit.ipynb". This model on the Mit_Bih dataset, achives an accuracy of 0.976. The training curves and confusion matrix are summarized in image 7.

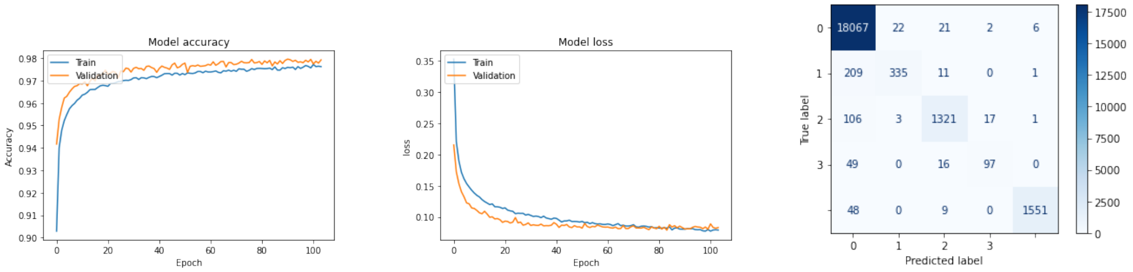


Figure 7: (left) train and validation loss of the transformer for the Mit_bih dataset.
(center) train and validation accuracy of the transformer for the Mit_bih dataset.
(right) confusion matrix of the transformer for the Mit_bih dataset

4.2 PTB

The code for this model can be found at "/src/PTB/Transformer_ptb.ipynb". This model on the PTB dataset, achives an accuracy of 0.968. The training, AUROC, AUPRC curves and confusion matrix can be found in images 8 and 9

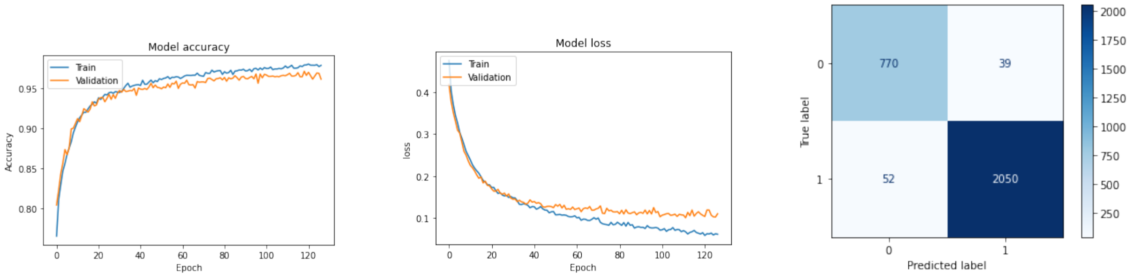


Figure 8: (left) train and validation loss of the transformer for the PTB dataset.
(center) train and validation accuracy of the transformer for the PTB dataset.
(right) confusion matrix of the transformer for the PTB dataset

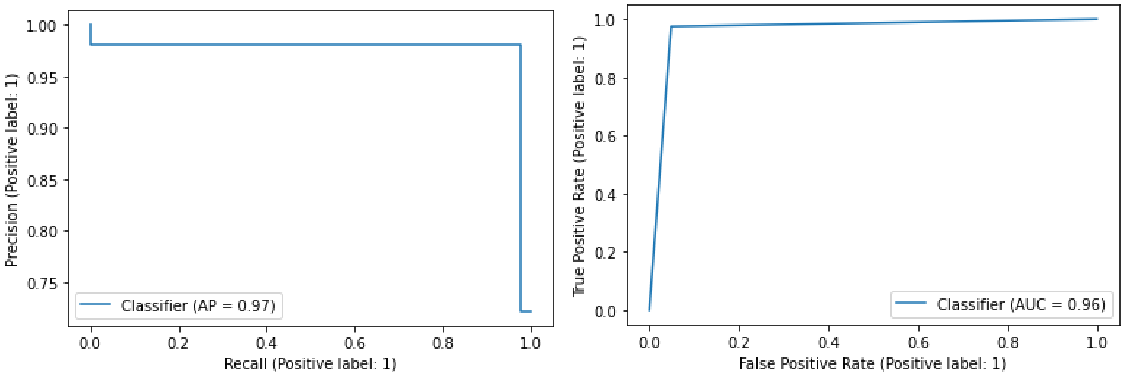


Figure 9: (left) AURPC curve of the transformer for the PTB dataset.
(right) AUROC curve of the transformer for the PTB dataset

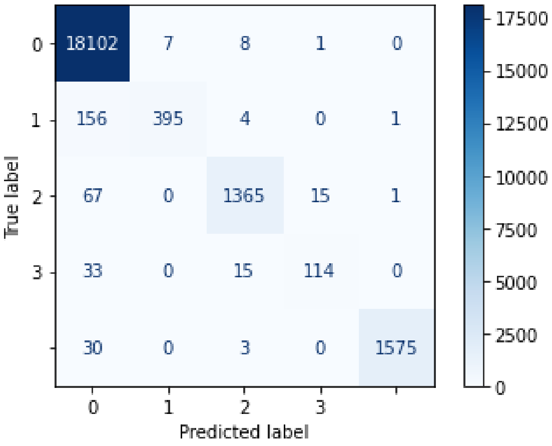
5 Ensemble of models

For this optional ensemble task, we decided to use two different approaches. For the first approach, we just summed up the different prediction probabilities of every model, and then we took the class with the highest one. For the second approach, we ran a logistic regression model on the output probabilities of every model, in order to better understand the relation between the models. The logistic regression ensemble achieves the best performance among all the models used for this assignment with an accuracy for the PTB dataset of 99.1%. Once again ensembles of models of different kinds prove to be more robust than each single model taken separately. The performance of each approach is summarized in their respective section.

5.1 Average of the outputs

5.1.1 MIT_BIH

The code for this ensemble model can be found at "src/MIT/ensemble_voting_mit.ipynb". This ensemble reaches an accuracy of 0.984. The confusion matrix can be found at 10.



True label \ Predicted label	0	1	2	3	
0	18102	7	8	1	0
1	156	395	4	0	1
2	67	0	1365	15	1
3	33	0	15	114	0
	30	0	3	0	1575

Figure 10: Confusion matrix of the ensemble average of the outputs for the Mit_Bih dataset

5.1.2 PTB

The code for this ensemble model can be found at "src/PTB/ensemble_voting_ptb.ipynb". This ensemble reaches an accuracy of 0.990. The AUROC, AUPRC curves and confusion matrix can be found in image 11

5.2 Logistic regression on the outputs

5.2.1 MIT_BIH

The code for this model can be found at "src/MIT/ensemble_logistic_regression_mit.ipynb". This ensemble reaches an accuracy of 0.987. The confusion matrix can be found at 12.

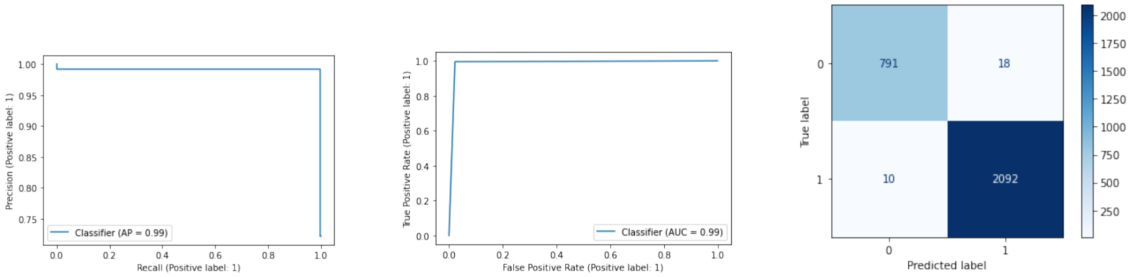


Figure 11: (left) AURPC curve of the ensemble average of the outputs for the PTB dataset.
(center) AUROC curve of the ensemble average of the outputs for the PTB dataset
(right) confusion matrix of the ensemble average of the outputs for the PTB dataset

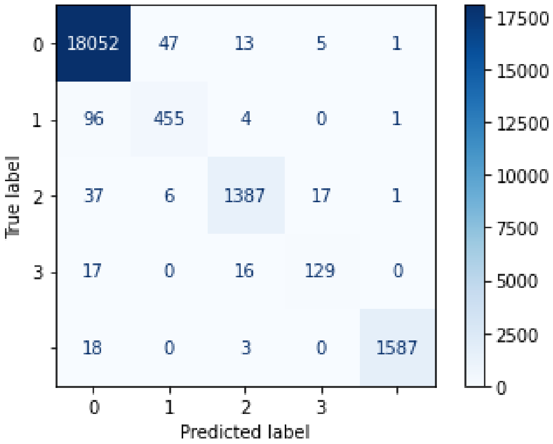


Figure 12: Confusion matrix of the ensemble logistic regression for the Mit_Bih dataset

5.2.2 PTB

The code for this ensemble model can be found at "src/PTB/ensemble_logistic_regression_ptb.ipynb". This ensemble reaches an accuracy of 0.991. The AUROC, AUPRC curves and confusion matrix can be found in image 13

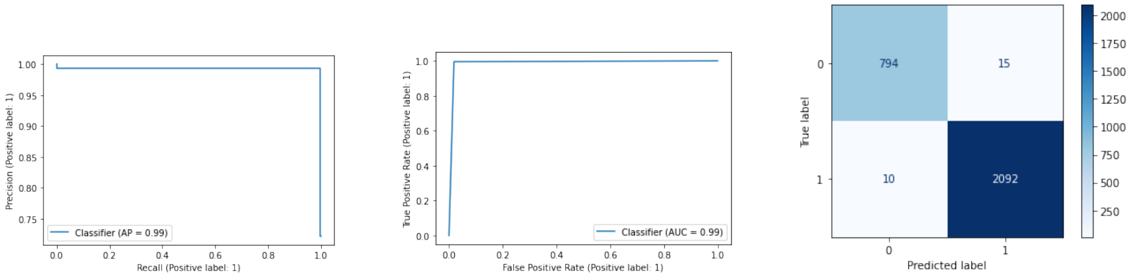


Figure 13: (left) AURPC curve of the ensemble logistic regression for the PTB dataset. (center) AUROC curve of the ensemble logistic regression for the PTB dataset (right) confusion matrix of the ensemble logistic regression for the PTB dataset

6 Appendix

Table 1: Summary of results

	MIT_BIH A	PTB
Vanilla CNN	0.981	0.982
CNN Residual Blocks	0.988	0.986
Transformer	0.976	0.968
Ensemble Average	0.984	0.990
Ensemble Logistic	0.987	0.991