ECG Heartbeat Classification

Phan Thanh Bình BI12-059

I. Introduction

The heart is the most important organ in the human body as it supplies and circulates blood, which every other organs needs to function. This vital role of the heart emphasizes the importance of monitoring its activity. One of the methods to assessing this activity by the use of Electrocardiogram (ECG), a non-invasive technique that records the heart's electrical signals. By analyzing these signals, healthcare professionals can gain valuable insights into the heart's health. However, manually analyzing ECG signals can be a labor-intensive and time-consuming task. With the advancements in technology, especially within the field of artificial intelligence, automating the analysis of ECG is totally feasible. In this work, we will:

- Explore the ECG Heartbeat Categorization Dataset.
- Build a deep learning model to perform the task required.

II. Dataset

According the author, the dataset comprises two sets of heartbeat signals extracted from well-known databases for heartbeat classification: MIT-BIH Arrhythmia and PTB Diagnostic ECG. We only consider the first one for our work which corresponds to $\frac{2}{4}$ downloaded files: *mitbih_train.csv* and *mitbih_test.csv*.

There are 87554 and 21892 samples in the train and test set respectively (109446 total). Each sample has the last column as its category and the first 187 columns as the signal sampling values. 5 different categories are available.

Figure 1 shows an example of a signal from the dataset. A large number of *zeros* are present at the end of the signal in order to make a consistent length with others.

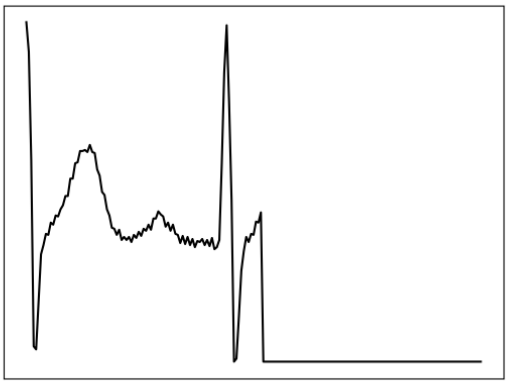

Fig. 1. Plot of an ECG signal using first 187 columns

Observing figure 2, we clearly see that the category distribution is highly imbalance in both the train and test set which will require some additional steps in preparing phase and metrics in evaluating phase.

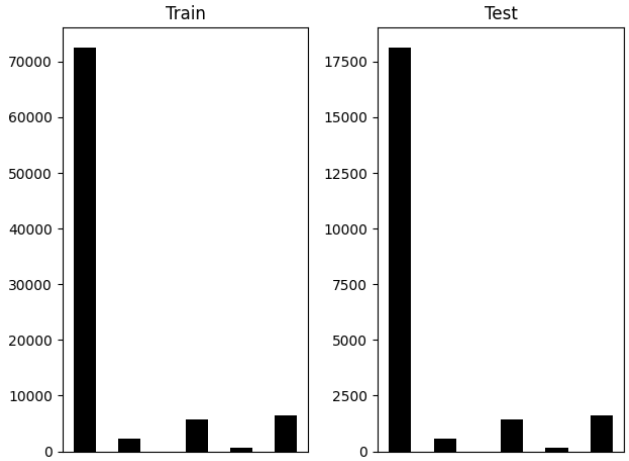

Fig. 2. Category distribution

III. Method

A. Preparation

One way for tackling imbalanced datasets is to oversample the minority class. The trivial approach would be to replicate instances from the each of those classes, however, this duplication fails to introduce fresh information to the model. Alternatively, new instances can be synthesized from the existing ones, a kind of data augmentation, commonly referred to as the *Synthetic Minority Oversampling Technique (SMOTE)*[1]. However, the excessive use of *SMOTE* can lead to undesirable outcomes since the generated data are based on hypothetical assumptions from existing data which can mislead the model. So from figure our detailed calculations for figure 2: $72471, 2223, 5788, 641, 6431$ signals for each category $0-4$ respectively, we decide to **undersample** category 0 to 7000 signals first, then use *SMOTE* to **oversample** other categories to 7000 signals each.

B. Model

Our model design is somewhat based on the architecture of VGG[2], detailed in figure 3. We employ the 1D convolution layers to extract the signal's features and maxpool layers to reduce the model complexity while still retain important features representations. Extracted features will then be flatten and passed through some fully connected layers. Finally, the

output will be a vector of 5 dimensions, represent the model's predicted score for the likelihood of each category. All hidden layers' sizes is chosen totally based on our preferences. The model is trained on 13 epochs with *cross entropy loss*, using *Adam optimizer* with *learning rate* 0.001 and *batch size* 64.

```
MyCNN(
  (extract): Sequential(
    (0): Conv1d(1, 64, kernel_size=(3,), stride=(1,))
    (1): BatchNorm1d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): MaxPool1d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (4): Conv1d(64, 64, kernel_size=(3,), stride=(1,))
    (5): BatchNorm1d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (6): ReLU()
    (7): MaxPool1d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (fc): Sequential(
    (0): Linear(in_features=2880, out_features=1024, bias=True)
    (1): ReLU()
    (2): Linear(in_features=1024, out_features=1024, bias=True)
    (3): ReLU()
    (4): Linear(in_features=1024, out_features=5, bias=True)
  )
)
```

Fig. 3. Model architecture

IV. Evaluation

A. Metric

Since this is a classification task, the accuracy of the model should be taken into account, however based on the skewed distribution of categories shown in figure 2, *accuracy* metric in general does not reflect well the performance of the model. For that reason, we analyze the confusion matrix of the model's predictions and calculate the *accuracy* for each category. A final number to show the model performance is the average of all the accuracies.

B. Result

Figure 4 show the confusion matrix. The model shows good accuracy towards categories 0, 2, 4 and decent towards others. Some explanations can be made about this result. Firstly, the model does not show absolute dominance in category 0, this can be the consequence of we undersample this category which limit the diversity of the data that the model can learn. Secondly, 2 classes 1, 3 get the relatively poor performance because they have least amount of *real data points*.

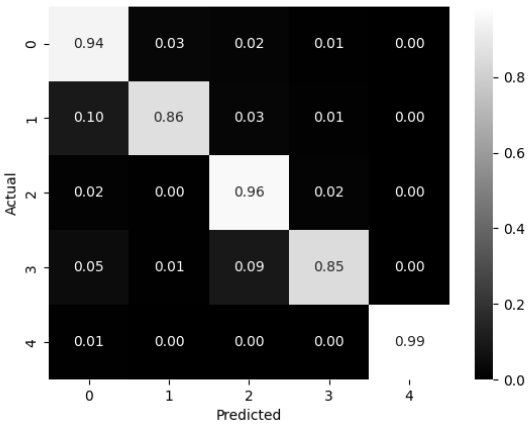

Fig. 4. Normalized confusion matrix of model's performance

V. Conclusion

In this work, we have explored a ECG Heartbeat dataset and build a simple deep learning model for classification task. The most difficult problem is handling the imbalance of the training and testing data which we have utilized both undersampling and oversampling techniques. The result shows the importance of having *real* data because it shows declined accuracies with undersampled categories and excessive oversampled ones.

References

[1] Nitesh V. Chawla et al. "SMOTE: Synthetic minority over-sampling technique". In: *Journal of Artificial Intelligence Research* 16 (2002). ISSN: 10769757. DOI: 10.1613/jair.953.

[2] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: 2015.