Classifications for ECG Readings

Timothy Fish
Virginia Polytechnic Institute and State University
timothyfish@vt.edu

Abstract

ECG readings are a primary measurement for diagnosing heart conditions in patients. ECG readings are used in a variety of scenarios in medicine. One of the most important scenarios is emergency medicine. ECG readings can quickly measure heart arrhythmias, so a doctor or paramedic can diagnose heart conditions. Especially in emergency medicine, a patient having a current heart attack must be treated immediately. These heart conditions need to be measured and diagnosed quickly. Currently, only doctors, nurses, or paramedics can diagnose based on an ECG reading. The goal of this paper is to see if technologies in machine learning could help diagnose heart conditions without the need for a certified professional. To approach this problem a dataset of heart arrhythmias was used to train a machine learning model. A one-dimensional convolutional neural network was used to train a model for five different types of arrhythmias. Experiments were run to test for model accuracy and loss. It was found after 25 epochs, the best model had an accuracy of 98.53% and a loss of 0.068. Overall the results were promising. However, when dealing with patients and medicine, accuracy needs to be extremely high. So further work must be done to obtain a model with a higher accuracy. More work needs to be done as well to classify more than five arrhythmias and determine an overall heart condition for the arrhythmia.

1. Introduction

This report aims to introduce a machine learning model to classify ECG readings for five different types of heart arrhythmia. The ECG readings were found through Kaggle and recorded by MIT and Beth Isreal Hospital in Boston[1]. The readings are classified into five different classifications: normal, supraventricular ectopic, ventricular ectopic, fusion, and unknown heartbeats. There was both a training and testing dataset given. In total, there were 109,446 samples in the training dataset, however, 87,553 samples were used to train the model. The testing dataset was used to validate and test the model. The datasets were broken up by 186 time steps with the last column representing its classification by a number ranging from 0 to 4. The previous columns represent measurements of the signal. A convolutional neural network was used to train the model. Throughout the testing of accuracies, multiple versions of the neural network were used. Throughout testing, metrics including loss and accuracy were checked to determine the performance of the model. However, accuracy was taken into account much more.

1.1. Background

Currently in medicine, a doctor, nurse, or paramedic can diagnose heart conditions off an ECG reading. ECG readings are also one of the most important tools in emergency medicine. ECG readings can record possible heart attacks and other life dangering conditions. That is why it is most important that someone can look at a patient's ECG and determine what steps need to be taken afterward.

There were various studies done to see if machine learning could be leveraged in the case of ECG readings. A paper written by Siti Nurmaini has a similar goal to classify ECG readings[2]. In their experiments, the model was trained using a one dimensional convolutional model. In their results, they found an accuracy of 99.98% for heart rhythms and a 99.87% accuracy for heartbeats. The difference in accuracies was mainly due to the complexity and individuality of heart rhythms and beats. The goal of this report is to have at least an equal accuracy of 99.98% and have the same accuracy for heart rhythms and heartbeats.

1.2. Motivation

In medicine, time is everything when treating a patient. Especially in emergency medicine, a patient is not necessarily stable and anything can happen in an instant. This is why ECG readings are one of the most important tools in emergency medicine.

With the limited amount of certified professionals that can diagnose from an ECG reading, it can be difficult to quickly determine the state of a patient. There is an evergrowing shortage of EMS personnel, specifically paramedics who can read ECG readings.

The hope is to allow for technologies with machine learning to diagnose and determine heart arrhythmias. Doing so could allow EMTs or anyone to gather more information about their patient or themselves. For instance, if a patient has a heart condition and there are no paramedics available, an EMT can take an ECG and with the trained machine learning model, they can communicate vital information to the hospital.

2. Approach

The initial approach had various parts to it. The first thing to do was to try and further understand the data. This included checking for data sparsity by organizing the training and testing datasets into different classifications. Plots of one example from each classification were created to try and understand the differences between each classification.

After data analysis, a one dimensional convolutional neural network was created for the model. The network had one layer and one stack. The model takes in a one dimensional signal and outputs a number for the classification. The model looks at measurements of the signal and determines what classification it falls into. The model was tested for accuracy and loss. After initial tests, it was found that accuracy was only 97.87%. The second approach involved adding another stack and another layer.

2.1 Data Analysis

The initial approach was to analyze the data by itself. At first, data sparsity was checked by organizing the training and testing datasets into their categories. Overall the data had good representation across all classifications.

Classification	Samples
0: Normal Beats	72470
1: Supraventricular Ectopic Beats	2223
2: Ventricular Ectopic Beats	5788
3: Fusion Beats	641
4: Unknown Beats	6431

Table 1: Number of samples for each classification in the training dataset

Classification	Samples
0: Normal Beats	18117
1: Supraventricular Ectopic Beats	556
2: Ventricular Ectopic Beats	1448
3: Fusion Beats	162
4: Unknown Beats	1608

Table 2: Number of samples for each classification in the testing dataset

To further understand the data, plots of each type of heartbeat were created. Once again, each heartbeat in each classification is not the same, and it will be a challenge to train a model to discern between the different classifications.

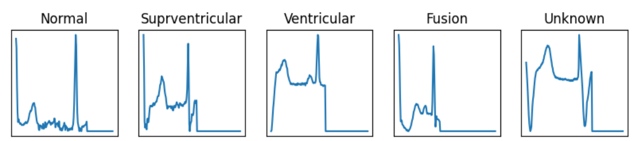

Figure 1: Example signals of a heartbeat for each classification

2.2 Model Creation

When creating a model, a convolutional neural network was chosen. This was chosen due to promising results from a similar project involving EEG readings for sleep stages[3]. The report stated an improvement in accuracy from previous experiments using a convolutional neural network. Furthermore, a one dimensional neural network architecture was used due to the data being one dimensional.

The first iteration of the model involved one stack with one layer of the convolutional neural network. The ReLU activation function was used for the layers, a filter of 64, and a kernel size of 3. Furthermore, one dimensional max pooling was added as well as a dropout of 0.25 after each stack. After the convolutional stacks were added, a dense layer with 64 units was added, followed by another dense layer with 5 units to ensure the output was the same as the number of classifications. A softmax activation function was added after the last dense layers. During model compilation, the adam optimizer and loss function sparse_categorical_crossentropy were used. 25 epochs were run to train the model This first approach worked well with a testing accuracy of 97.87%. However, a much higher accuracy was preferred.

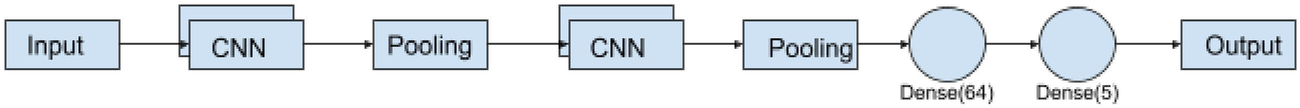

Figure 2: Drawing overview of the second and final model

The second iteration involved adding more layers and more stacks. An additional stack was added as well as an additional layer in each stack. In this model, two stacks with two layers were used. A pooling layer was added in between the CNN stacks. Again two dense layers were added at the end one with 64 units and the last with 5 units to ensure output was the same as the number of classifications. All other implementations and model parameters remained the same. With the modifications, there was a much higher accuracy in the testing dataset. The second iteration involved an adaptation from a model given in a previous class assignment[4]. This was a novel approach due to the addition of multiple layers and multiple stacks. Other approaches involved using one stack with multiple layers, or multiple stacks with one layer.

2.3 Problems to Approach

The first approach to this problem had promise but could be improved upon. With the first model iteration having an accuracy of 97.87%, it did not match previous findings of 99.98%. In this case, the second approach was done to further improve accuracy.

The second case saw an improvement in accuracy at 98.53%. However, this still does not match previous findings. Moreso, there could be problems with the possible addition of more classifications. It is believed that more epochs will not fix this issue as accuracy starts to level out at 9 epochs and loss does not decrease after 10 epochs.

More layers and stacks were later added however resources were exhausted with an extended runtime. It was anticipated that accuracy would have increased with further layers and stacks.

3. Experiments and Results

Using the given training data, the model was trained over 25 epochs. The model was also tested for loss and accuracy over the given testing data. After each epoch, loss, and accuracy for both training and validation were saved to later plot.

3.1 Experiment

An initial experiment was run with the first approach to the model. The experiments primarily looked at loss and accuracy over the number of epochs. The testing dataset was used for validation. Loss and accuracy were primarily looked at in measuring performance on the model. In the first experiment, accuracy was increasing up until epoch 5. Loss was decreasing also up until epoch 6. Overall, the best model of this iteration had an accuracy of 97.87%. With the goal being in mind to have accuracy at least to 99% the second iteration of the model was created.

The second model iteration had the same experiment parameters as the first. The second iteration had a higher accuracy of 98.53%, but again this did not reach the target accuracy.

A third and final experiment was run with a third model, with an additional stack and layer. However, this experiment was not complete as the runtime was much longer than the first two experiments and as a result, resources were exhausted.

3.2 Results

Through the second experiment, there was a slightly higher accuracy. Accuracy also stayed increasing for longer until epoch 9 compared to the first experiment with accuracy stagnating at epoch 5.

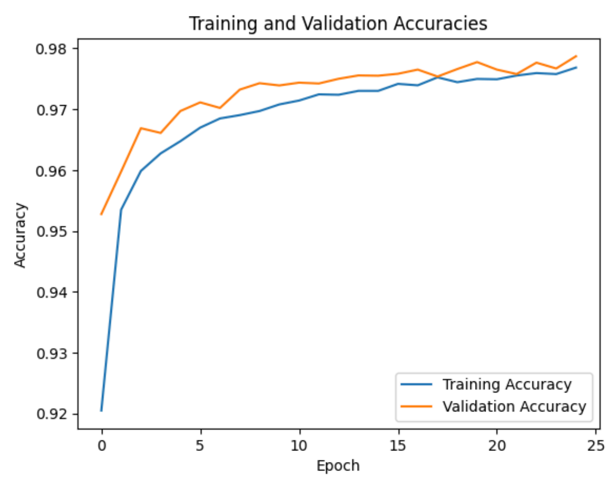

Figure 3: Training and validation accuracies for the first iteration of the model

Figure 4 Training and validation accuracies for the second iteration of the model

Interestingly, for the first iteration, validation accuracy finishes with higher accuracy than the training accuracy. With the first model, it is shown that overfitting is not an issue. Accuracy does become stagnant after the fifth epoch. Loss becomes stagnant after the sixth epoch.

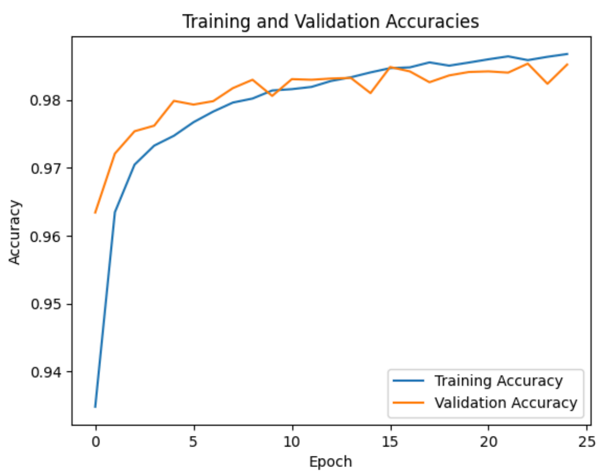

Figure 5: Training and validation accuracies for the second iteration of the model

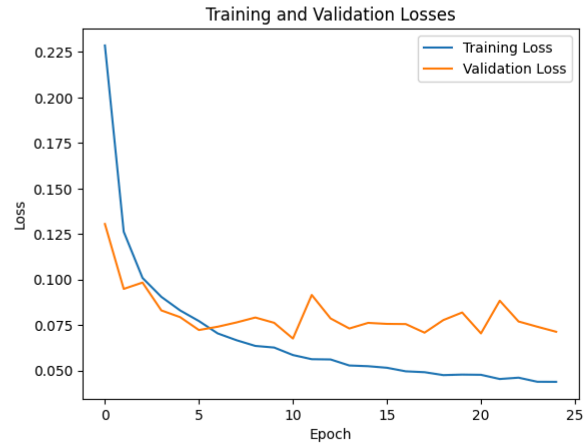

Figure 6: Training and validation loss values for the second iteration of the model

In this second iteration, training and validation accuracies slightly increased. Losses were also less for both training and validation in the second iteration of the model.

Quantitatively a table is given showing the loss, and accuracy for the training and validation datasets for both the first and second iterations of the model.

	Training	Validation
Loss	0.079	0.097
Accuracy	97.68%	97.87%

Table 3: Loss and accuracy for the first iteration of the model

	Training	Validation
Loss	0.042	0.068
Accuracy	98.68%	98.53%

Table 4: Loss and accuracy for the first iteration of the model

In the first iteration validation accuracy was higher than the training accuracy. This is a sign that the model does not overfit the data and generalizes the data. However, in the second iteration, validation has a lower accuracy than the training accuracy. Although this may be a sign of overfitting, the accuracies are close to each other so this should not be the case.

In the second iteration, both accuracies and losses see an improvement. Although the second model may be overfitting more than the first, the higher accuracy and lower losses show a better performance for the second model. Overall, the second model has better performance and generalizes well.

The second model, however, did not reach the goal accuracy of 99.9%. It is believed that the lower layer count may be the reason for this. However, when experiments were run with a model with a higher layer count, runtimes were much longer. As a result, resources were expended and results were not found.

4. Availability

The code written for this report is available freely for anyone's use through GitHub[5] as well as both iterations of the trained model. This report with the findings will also be available on GitHub. Data files for the project however will not be provided on GitHub as the files are too large to upload. Instead data files could be found on Kaggle[1]. Specifically, the files mitbih_train.csv and mitbih_test.csv were used for this project.

5. Reproducibility

The model was trained with TensorFlow and Keras. All model parameters can be loaded by anyone with the code. Once again the data is not readily provided through GitHub, as they are too large, but they can be found on Kaggle.

Model parameters are completely reproducible as they are parameters provided by TensorFlow. Results may vary slightly for someone else running the project. However, accuracies should be within 0.1% and loss should be within 0.02 for the given findings.

REFERENCES

[1] "ECG Heartbeat Categorization Dataset," *www.kaggle.com*. https://www.kaggle.com/datasets/shayanfazeli/heartbeat?resource=download (accessed May 01, 2024).

[2] A. Darmawahyuni *et al.*, "Deep learning-based electrocardiogram rhythm and beat features for heart abnormality classification," *PeerJ Computer Science*, vol. 8, p. e825, Jan. 2022, doi: https://doi.org/10.7717/peerj-cs.825.

[3] A. Supratak, H. Dong, C. Wu, and Y. Guo, "DeepSleepNet: A Model for Automatic Sleep Stage Scoring Based on Raw Single-Channel EEG," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 11, pp. 1998–2008, Nov. 2017, doi: https://doi.org/10.1109/tnsre.2017.2721116.

[4] https://canvas.vt.edu/courses/185672; Homework assignment 4

[5] Timothy Fish, "tfish22/ML_ECG," *GitHub*, May 01, 2024. https://github.com/tfish22/ML_ECG (accessed May 01, 2024).