ECG Classification

**Objective**

Our project topic covers the analysis of ECG’s (electrocardiogram) of patient heartbeats to classify heart conditions. There are five classification categories. Zero denotes normal case and one through four denotes a certain type of heart abnormality. Each instance within our dataset represents an “image” of a patient’s ECG over a 187 millisecond time interval where each millisecond is represented with a normalized data point that captures the ECG value at that given point in time. Our first objective is to use/test several classification models to find which models provide the highest overall accuracy. The study which we obtained the data used a CNN (convoluted neural network) model <https://arxiv.org/pdf/1805.00794.pdf>, however we wanted to know if a RNN (recurrent neural network) could produce better results since the data is time based. We also want to see if we can reproduce the original study’s results with CNN.

Our second objective was to evaluate other techniques which could improve performance of our neural network models. These were data augmentation and a two stage NN (neural network) architecture. Regarding the two stage NN, one stage was responsible for just classifying whether the ECG signal was normal or abnormal. This model and its weights were saved and frozen then fed into another NN which then classified the specific heart abnormality, categories one through four. Through specialization of the tasks we hoped to optimize the NN design. The reason why we chose these techniques is because the data set was unbalanced; the abnormal sample sizes were quite low compared to the normal samples. Through data augmentation we could increase sample size by synthesizing data by injecting noise into the original signal and thereby balancing the sample sizes in each category.

**Dataset Description**

As stated above, each instance within our dataset represented a single recorded ECG signal. The training and validation data consisted of 87,553 and … observations, respectively. The individual data points of each instance were normalized values between 0 & 1. These represented the feature values which totaled 187. Most signals did not have a value for each of the 187 features and was padded with zeros. A sample of the ECG signal is shown below in Fig1. The data set was obtained through Kaggle web site <https://www.kaggle.com/shayanfazeli/heartbeat>.

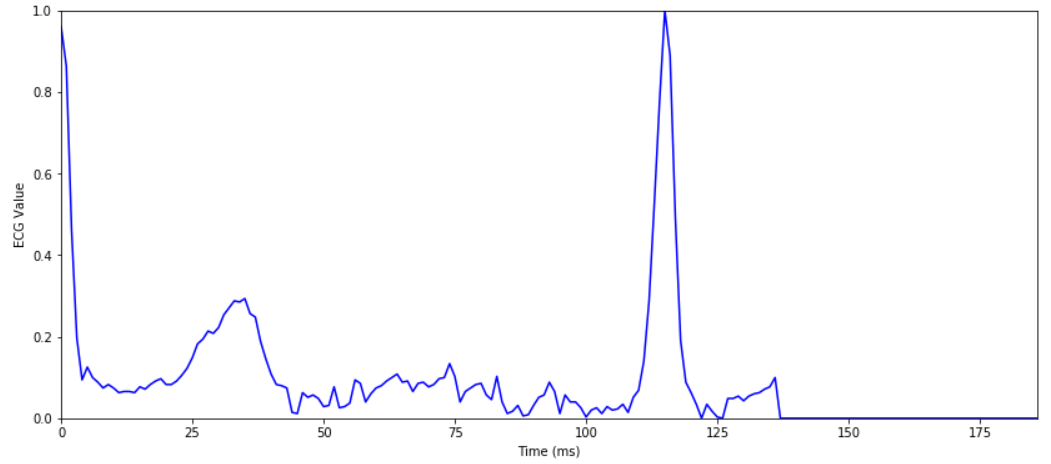


Fig. 1 Sample of ECG signal

The histogram below shows the training set labels, categories 0-5. There are a much larger number of normal samples compared to the abnormal samples. In the training set there were over 70,000 observations for the normal category and as few as 641 in the abnormal category. This can cause problems with machine learning algorithms…(details). Categories two and three have a particularly a low count being less than 2.5% of the total.

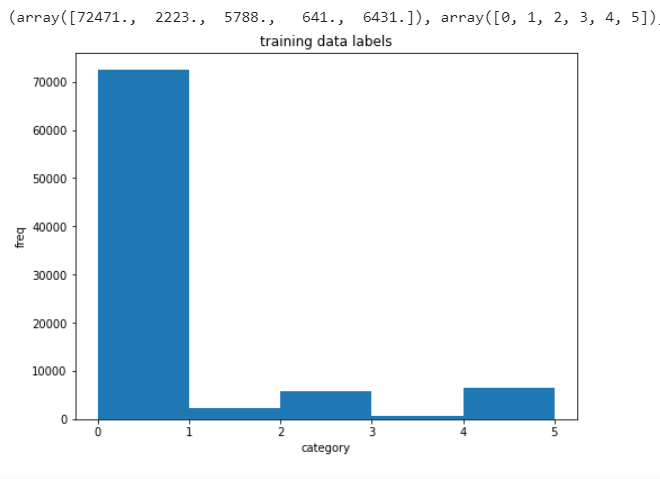


Fig. 2 Histogram of training data labels

**EDA/preprocessing**

Add details ….

A close up of a map

Description automatically generated

A close up of a map

Description automatically generated

A close up of a map

Description automatically generatedA close up of a map

Description automatically generated

A close up of a map

Description automatically generatedA close up of a map

Description automatically generated

**Data Augmentation Description**

To improve the balance of data between normal and abnormal categories we augmented the abnormal data sets by injecting noise into its data. The normal training data set had up to 70,000 samples while the abnormal samples had a low of 641samples. A function called createnoise() was created to perform this task. From the four abnormal categories this function randomly injected a certain amount of noise into the signal. The figure below shows one signal with max noise at 10%. However, for the study we chose to use 6% as to not distort the signal too much. In the end, we balanced the data set so that each abnormal category would have a sample size of n=10,000. Furthermore, when we fit the data to our model, we also included 10,000 samples of the normal category instead of the 70,000. Thus, for this new data set the total sample size was 50,000 for the five total categories, 0-5. Data augmentation was only performed in the training data set. The validation data set was left untouched.

A screenshot of a cell phone

Description automatically generated

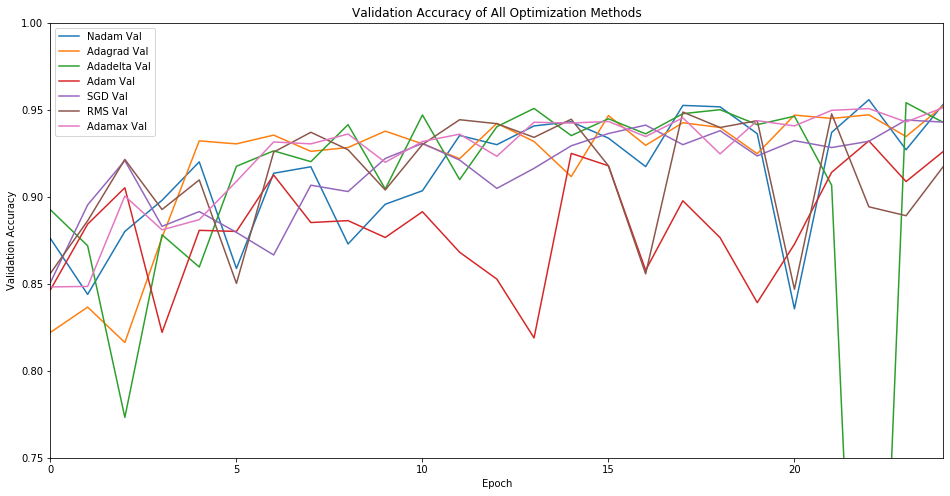
Fig. x ECG signal with (upper plot) and without (lower plot) data augmentation

Models

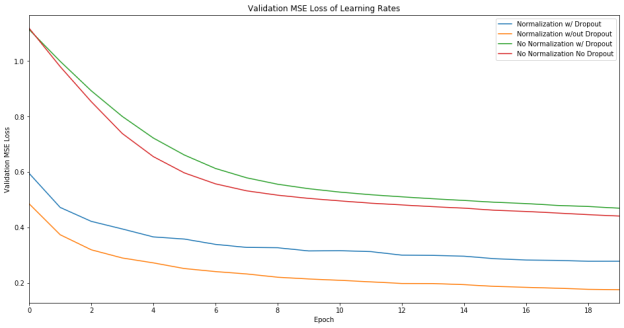
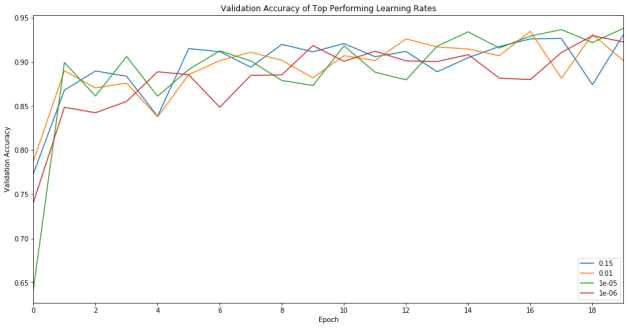
CNN

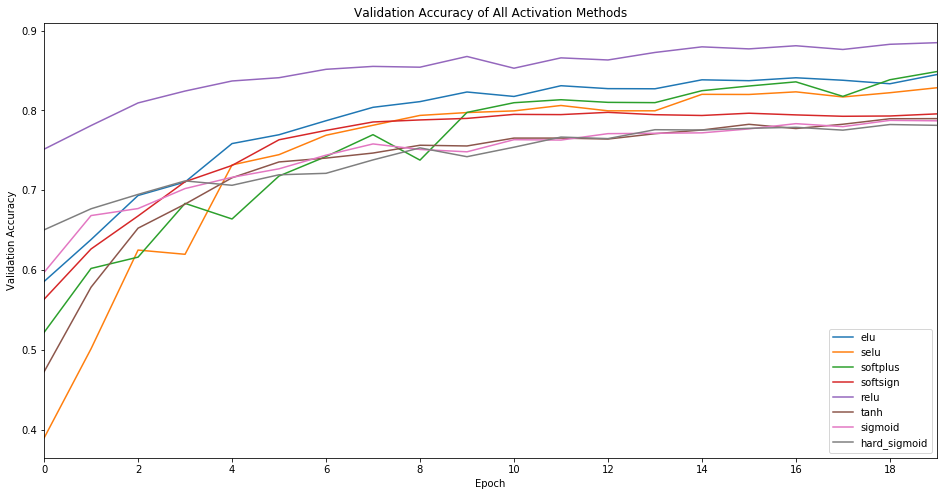
The architecture of our CNN is pretty straight forward: Two 1-D convolution layers, a 1-D pooling layer, and two dense layers (we also normalize the inputs between each layers). We experimented with other archetypes/layouts but found that this gave the best results of all our attempts.

Once we established the layout for the CNN, we set out to find the optimal optimizing method. We tested the following methods: Nadam, Adagrad, Adadelta, Adam, SGD, RMS, and Adamax.



Of all the tested methods, Adamax gave the best performance (with the highest accuracy, lowest loss, and most stable learning session). From there, we looked towards the model’s learning rate, activation method, and whether or not to include our normalizing/dropout layers in our model in an attempt to maximize the model’s performance. After all our tests, we found that a learning rate of 1e-5, RELU activation method, inclusion of normalizing layers, and exclusion of the dropout layers gave the best performance. The graphics for all of this are included below.

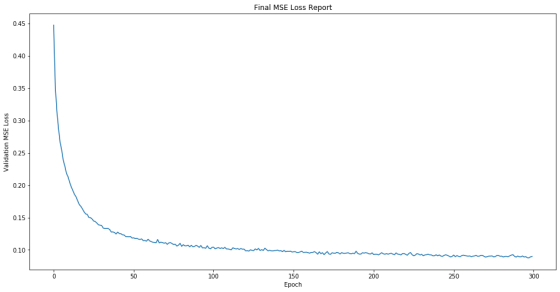
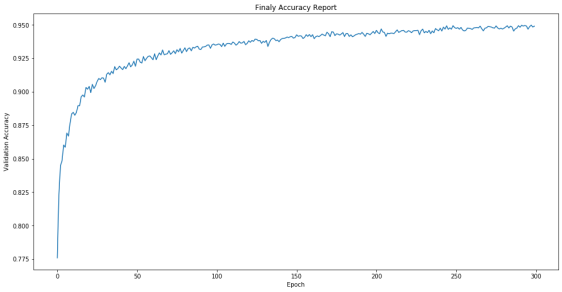




The final results of the CNN are:

% Validation accuracy: 95%

MSE Loss: 0.04



RNN

Summary

Video link

Kaggle source: <https://www.kaggle.com/shayanfazeli/heartbeat>

Other source: <https://arxiv.org/pdf/1805.00794.pdf>