CS230

Detecting Termination of Atrial Fibrillation Events on Electrocardiograms Using Deep Learning

Healthcare

Nik Caryotakis
nikcaryo@stanford.edu

Cortney Weintz
cweintz@stanford.edu

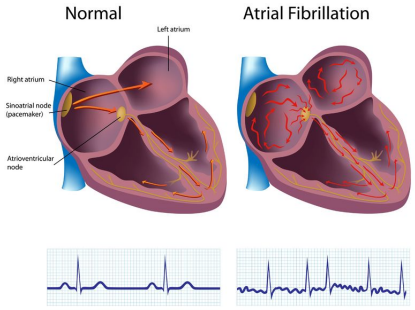
Eric Xu
ericxu24@stanford.edu

1 Abstract

Atrial fibrillation (AF) is a quivering or irregular heartbeat (arrhythmia) that can lead to blood clots, stroke, heart failure and other heart-related complications. To better understand the condition, we employed deep learning nerual network methods (CNNS, RNNs, LSTM, etc.) to try and identify the end points of AF episodes. Although our results were inconclusive for now, further modifications seem promising.

2 Problem Statement

The American Heart Association (AHA) estimate that 2.7 million American people live with some form of AF.



Early detection of AF is of great value for surgery options, drug intervention and diagnosis. Unfortunately, there is currently no algorithm that can efficiently measure the onset and end of AF episodes. Previous AF detection algorithms usually focus on the classification of different types AF instead of locating the onsets and ends of AF episodes [1]. Instead of focusing on classification, in our project we aim to expand the understanding of AF by trying to detect the end of AF events. We hope that our insights can extend to the broader understanding of AF.

CS230: Deep Learning, Fall 2021, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

3 Dataset

The dataset we used comes from the State Key Laboratory of Bioelectronics, Southeast University, China. Data are recorded from 2-lead wearable electrocardiogram (ECG) monitoring devices, each sampled at 200 Hz [1]. In order to avoid ambiguity in annotation, an AF episode is limited to contain no less than 5 heart beats [1].

In total, we had 1196 training examples of which 475 were AF examples and 721 were normal, non-AF, ECG examples [1]. (The included datum examples are only showing data from 1 of the leads for readability.) The test set was not released by the lab. Thus, we will construct our own test set by extracting a subset from the provided training data.

All data is provided in WFDB format and the annotations are standardized according to PhysioBank Annotations. The annotation includes the beat annotations (R peak location and beat type), the rhythm annotations (rhythm change flag and rhythm type) and the diagnosis of the global rhythm.

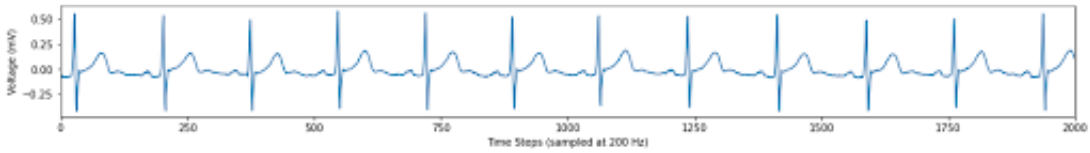


Figure 1: *Example of a non-AF example.*

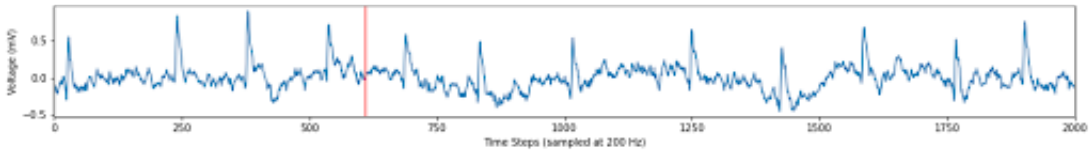


Figure 2: *Example of an AF example. (Note that the red line indicates the termination of the AF event.)*

However, we should note that there were some examples that didn't pass a sanity check.

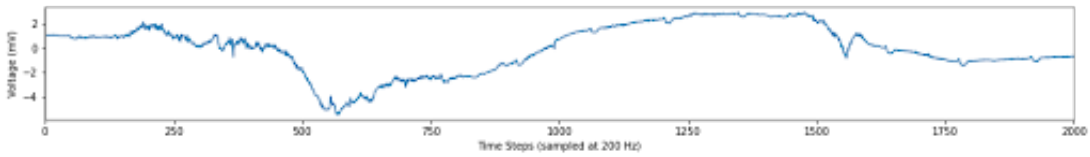


Figure 3: *Flawed non-AF example.*

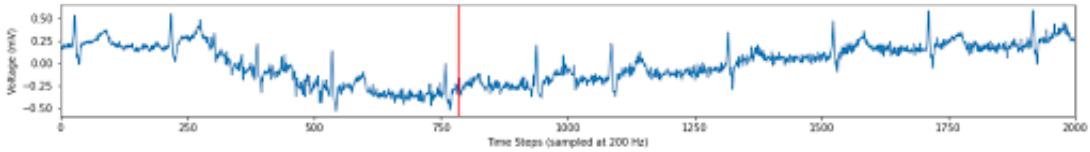


Figure 4: *Flawed AF example.*

We split the training data into train/val/test set splits of 70/20/10, trimmed all examples to 15000 time steps, and zero-padded the ones that were shorter to 15000. We also labeled the data so that all time steps of non-AF examples were 0's and that only the 50 time steps after the AF event ended were labeled 1's.

4 Methods

All experiments were conducted on an Amazon EC2 p2.xlarge instance (1 GPU, 4 CPUs, 61 GB RAM).

Our initial model was inspired by the Qihang Yao paper (12 lead ECG, 1D CNNs + LSTM + Attention, classified entire ECG) as well as the hot word detection example from class [2]. Several convolutional layers enable the spatial fusion of information from different ECG leads. Following the convolutions layers were two LSTM cells and a time-distributed fully connected layer with sigmoid activation [2]. For the loss function, we use binary cross-entropy loss. Intuitively, we hypothesized that the CNN can extract spatial features, and the LSTM will learn temporal relationships.

We also built a model based on the Hannun paper [3]. In this setting, rather than predicting the end of an AF event as marked by a series of 1's, we encode the input vector (2-lead ECG) into a smaller 1D state space and classify each entry as either AF (1) or non-AF (0) [3]. To do this, we use a deeper network of 1D CNNs than in the Yao-inspired model and forgo the LSTM cells. For the loss function, we use binary cross-entropy loss.

5 Results & Discussion

Initially, chose the smaller model inpsired by the Yao and employed it to a trigger-word detection setting (0000011111000) [2]. This paper had success with 1 dimensional CNNs, and no preprocessing or Fourier transforms or spectrograms was necessary.

To test the validity of this architecture, we initially trained it on a small number of examples ($n = 50$), with a learning rate of 1^{-4} for 50 epochs. Some of the results on the validation set can be seen below. The trained model had low loss (<0.04) on both the training and validation set within 5 epochs.

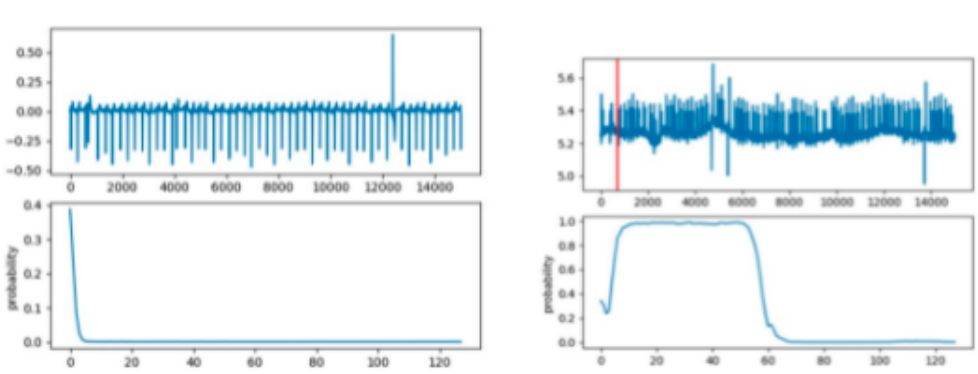


Figure 5: *Examples of non-AF and AF, respectively, with corresponding probability of AF classification calculated from our model.*

Notice that the Yao-inspired model outputs high probabilities for around 50 timesteps after the end of the AF event, as expected. However, in viewing these results, we noticed a trend that our positive AF examples all had mean signals > 0, while the negative examples had mean signals near 0. After we normalized all input data (subtracting the mean) and retraining on the same number of examples, we could not reproduce this result. We then trained on the full dataset, normalized, over 100 epochs and at learning rates of 1^{-3}, 1^{-4}, and 1^{-5}. The $LR = 1^{-4}$ model was the only one that successfully fit the training data (none fit the validation data), and we encountered a new problem with the predictions. All predictions were essentially identical and looked like the images below.

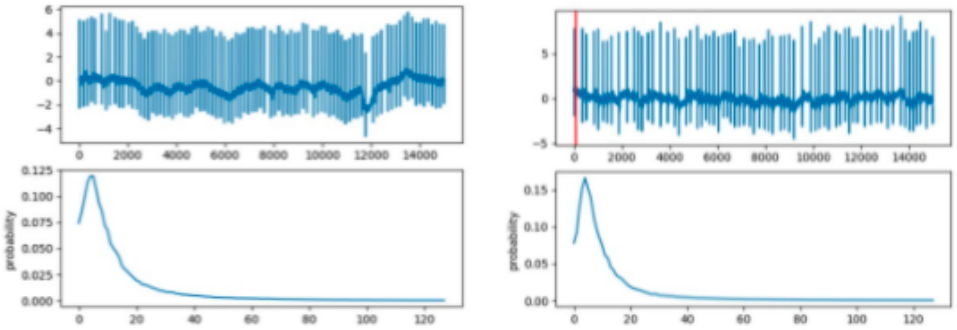


Figure 6: *Non-AF and AF examples, respectively, with corresponding probability of AF classification calculated from our "big" model.*

This generally matches the distribution of AF end points in the training data, so this is a learned 'magic' prediction of sorts that achieves low loss on all of the training data.

Seeing these results, we conducted additional experiments. Firstly, we tested various hyper-parameters, specifically, a 1000-length output vector with 50 1's to signal end, a 128-length output vector with 50 1's to signal end, and a 128-length output vector with 5 1's to signal end.

Nonetheless, each one produced the same sort of outputs suggesting the learned 'magic' prediction. Thus, we hypothesized that the hot-word detection framing was not suitable to this task and sought to emulate the Hannun paper/setting instead [3].

For our Hannun-inspired model, we encoded the original 2-lead ECG into a smaller vector, with each entry corresponding to the probability that the respective ECG sequence is part of an AF event. We also reduced the number of convolutional layers, as we had significantly less data than Hannun, but otherwise it is the same. We trained for 100 epochs with a batch size of 50 at three different learning rates; the results can be seen below.

Learning Rate	Train Loss	Val Loss
1e-3	0.023	0.856
1e-4	0.075	0.884
1e-5	0.18	0.956

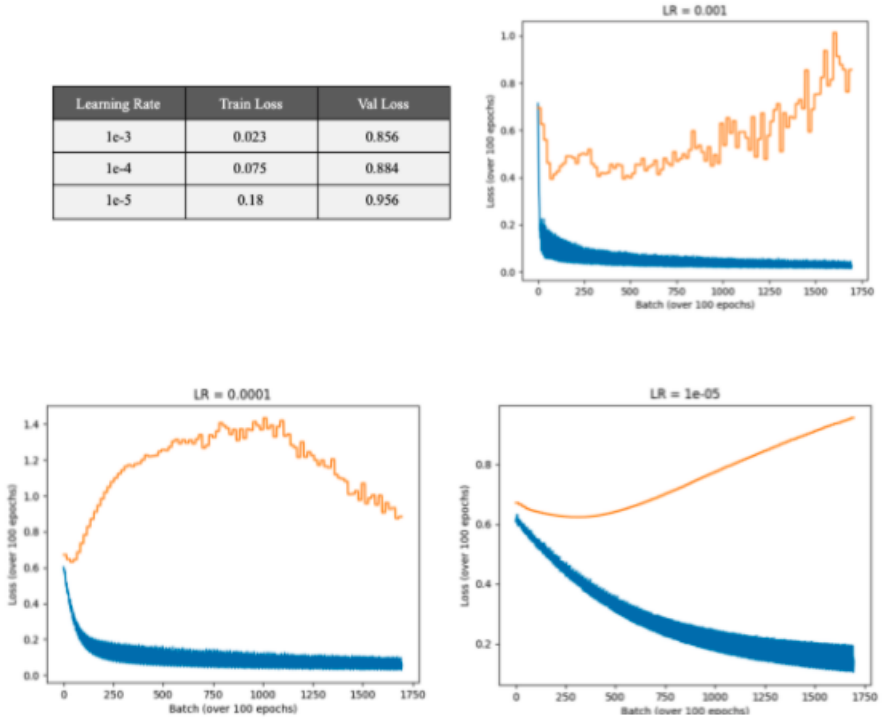


Figure 7: *Graphs of Hannun-inspired model's validation loss (orange) and train loss (blue) for the three learning rate options.*

As we can see, all models overfit. Some predictions from the $LR = 1^{-3}$ model on the validation set can be seen below.

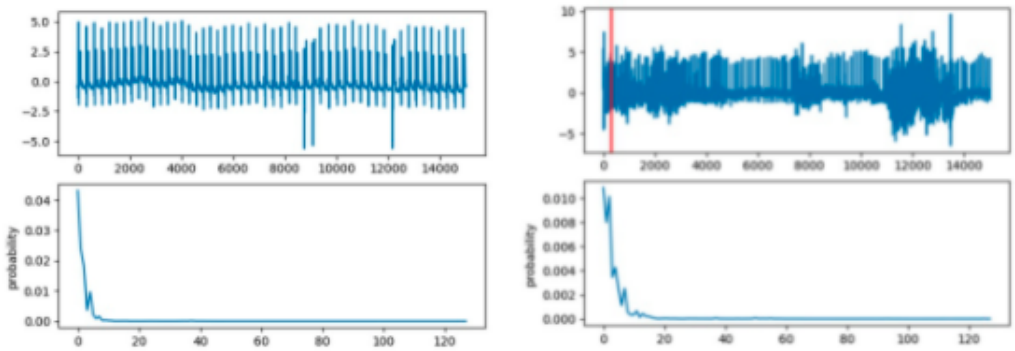


Figure 8: *Sample non-Af and AF examples, respectively, from the "big" model with* $LR = 1^{-3}$.

As we can see from the examples, we're encountering the same issue as in the trigger-word setting. The model learns a function that fits the training data well, which essentially results in the same prediction every time. When analyzing the probability plots, there seems to be a distribution that roughly aligns with where AF events usually end.

Since we weren't confident in any of these models, we refrained from running our model on the test set and are holding it out for future work.

6 Conclusion & Future Directions

Unfortunately, we were not able to achieve interesting results given the clear flaws in our approach. However, we still believe that within our original research problem statement are valuable insights that can be feasibly found.

To improve our likelihood of success, in the future we would collect additional data from more external databases (e.g. the one from the Hannun paper), then up/down sample accordingly. We would also have to dedicate more time to properly clean and normalize our data. Consulting someone with domain expertise in ECG recordings may aid this process.

Alternatively, we could apply transfer learning with the Yao-inspired model and expand our scope of research to analyze Paroxysmal-AF as well. In the original paper, a single ECG recording was split into 1s intervals where each one was classified as 1 of several types of rhythms. Presumably, we could simply add the Paroxysmal-AF as an additional class and retrain the last couple layers. This is the direction we find most promising in terms of immediate next steps.

Lastly, as all of our models exhibited overfitting, we considered exploring data augmentation procedures specific to ECG recordings. However, the Nonaka paper shows that this only improves performance by a few percentage points and therefore it would likely not have helped with the issues we encountered. [4].

7 Contributions

All team members contributed equally to the research and design aspect of the project. Nik handled AWS setup. Code was worked on collaboratively in a colab notebook before porting to local machines and then AWS. The report was completed equally. Cortney and Eric produced the final video.

Github and Code

https://github.com/nikcaryo/cs230-ecg-afib

References

[1] Wang, X., Ma, C., Zhang, X., Gao, H., Clifford, G., & Liu, C. (2021). Paroxysmal Atrial Fibrillation Events Detection from Dynamic ECG Recordings: The 4th China Physiological Signal Challenge 2021 (version 1.0.0). PhysioNet. https://doi.org/10.13026/ksya-qw89.

[2] Qihang Yao, Ruxin Wang, Xiaomao Fan, Jikui Liu, Ye Li, Multi-class Arrhythmia detection from 12-lead varied-length ECG using Attention-based Time-Incremental Convolutional Neural Network, Information Fusion, Volume 53, 2020, Pages 174-182, ISSN 1566-2535, https://doi.org/10.1016/j.inffus.2019.06.024.

[3] Hannun AY, Rajpurkar P, Haghpanahi M, Tison GH, Bourn C, Turakhia MP, Ng AY. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. Nat Med. 2019 Jan;25(1):65-69. doi: 10.1038/s41591-018-0268-3. Epub 2019 Jan 7. Erratum in: Nat Med. 2019 Mar;25(3):530. PMID: 30617320; PMCID: PMC6784839.

[4] Nonaka, Naoki, and Jun Seita. "Data Augmentation for Electrocardiogram Classification with Deep Neural Network." The 35th Annual Conference of the Japanese Society for Artificial Intelligence, 2020.

[5] Piyush Jain, Pranjali Gajbhiye, R.K. Tripathy, U. Rajendra Acharya, A two-stage deep CNN architecture for the classification of low-risk and high-risk hypertension classes using multi-lead ECG signals, Informatics in Medicine Unlocked, Volume 21, 2020, 100479, ISSN 2352-9148, https://doi.org/10.1016/j.imu.2020.100479.

[6] Weimann, K., Conrad, T.O.F. Transfer learning for ECG classification. Sci Rep 11, 5251 (2021). https://doi.org/10.1038/s41598-021-84374-8