Long Short-Term Memory Neural Networks for Sequence Classification and Prediction of Cardiac Arrest Events

JONATHAN KING

Department of Bioengineering, University of California at Berkeley
Department of Physiological Nursing, University of California at San Francisco

ABSTRACT—Long Short-Term Memory(LSTM) neural networks are explored as a method of classifying sequences of hospital alarms to predict cardiac arrest events. Current performance (AUC = 0.81) suggests the LSTM model can at least match performance of existing methods [1]. It is expected that augmenting the dataset to increase the sample size in addition to building a more complex evaluation engine should provide an increase in model performance. These improvements and others will be completed Summer 2017.

I. BACKGROUND

Bedside monitors in the hospital setting are essential to the work of caregivers by delivering important information about a patient's health in a timely fashion. Their effectiveness relies on their sensitivity as well as the ability of human actors to interpret their signals and intervene when necessary. Unfortunately, the tremendous volume of alarms in modern clinics can lead to alarm fatigue, in which clinicians become desensitized to the constant barrage of audible alarms and end up missing actionable opportunities for treatment. A past study at UCSF [2] found that in one Intensive Care Unit (ICU), each hospital bed produced 187 alarms per day. Another important finding was that 88.8% of the alarms for heart arrhythmias in the study were false positives.

Recent efforts by the Hu Lab at UCSF have been aimed at extracting the most relevant bedside alarms and constructing SuperAlarm patterns, robust subsets of alarms that have shown to be better predictors of patient deterioration [3]. In particular, the Hu Lab has focused on techniques to predict cardiac arrest (a.k.a code blue) events using SuperAlarms to encode the raw alarm data. Work by Rebeca Salas-Boni in 2015 [4] showed that by generating a time-series model to represent sequences of SuperAlarm events over time, logistic regression could be used to predict code blue events with up to 90.9% accuracy. Following such good performance, the lab endeavors to increase sensitivity and specificity to the point that a predictive algorithm could be used in practice.

II. PROJECT MOTIVATION

In recent years, deep neural networks have been the subject of much interest across disciplines, largely due to their ability to efficiently solve many complex image and audio classification tasks [5]. Recurrent Neural Networks (RNNs) are neural networks that retain some amount of information between each iteration (**Fig 1**). Because they take into account the temporal relationship between events, RNNs are particularly well-suited for sequence classification tasks where the probability of an event is dependent on the ordering of prior events.

One particularly popular implementation of a RNN is the Long Short-Term Memory network (which is able to determine exactly how much information is kept in the model's hidden state between iterations. This added complexity has contributed to their success and, as a result, LSTMs have been adopted by many as the de facto neural network for sequence classification tasks.

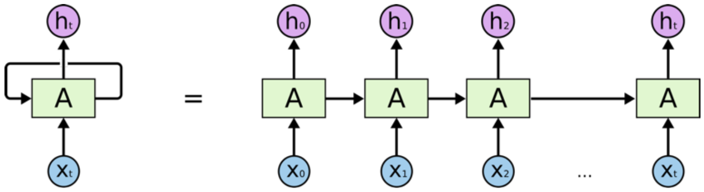


Figure 1: *Simplified RNN architecture.* [6]

Such advancements in the field of sequence classification using LSTMs have motivated this work to improve upon the accuracy of existing code blue prediction models. Many alternative approaches have relied on aggregating alarms within a set window of time without respect to their precise ordering. Due to the natural ability of LSTMs to analyze sequential information without aggregation, we expect the performance of an LSTM model on our alarm dataset to match, if not exceed, existing approaches.

III. Patient Data

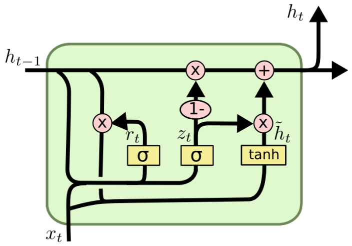
The data for this project comes from ICUs in two hospitals: The UCLA Ronald Regan Medical Center in Los Angeles and the UCSF Parnassus Medical Center in San Francisco. For the purposes of this project, only the sequence of monitor alarms will be used to train the model and the rest of the data (i.e. EEG, ICP waveforms, etc.) will be ignored. For training the model, patients are classified as either "Control" or "Code Blue" and the patient's full sequence of integer alarm codes is fed into the LSTM network as a series of one-hot vectors. On the model's first training and evaluation, data from 300 code blue and control patients has been included.

After gaining access to more data, the model is expected to train on around 500 code blue and control patients. Furthermore, in order to decrease the data's over-representation of control patients, the next steps of the project will include sampling many n-hour alarm subsequences from each code blue patient, thereby increasing the sample size from the code blue population.

IV. Implementation

The Python library Keras has been a popular choice for deep learning models because it contains a relatively high-level API that can run on TensorFlow or Theano, two of the most widely-used machine learning libraries for Python. Its syntax enables quick creation and validation of models. For this project, the full LSTM network was constructed with the following layers:

1. Dropout(0.2); designed to prevent overfitting, this layer ignores 20% of the input units, preventing the network from co-adapting hidden units.
2. LSTM, with the following architecture;



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Figure 2: *LSTM architecture.* [6]

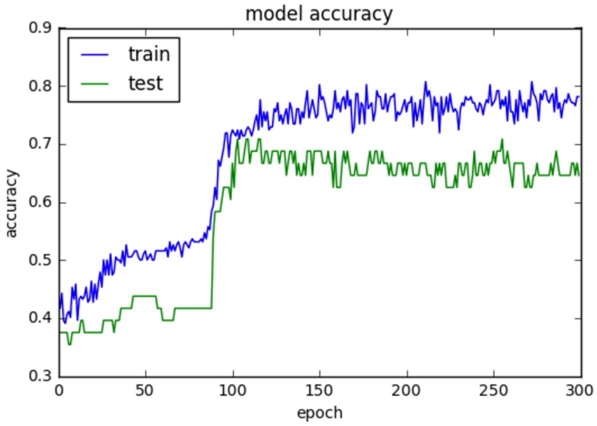
3. Dropout(0.2); added once more before the output layer to prevent overfitting.
4. Sigmoid; this ensures the output falls between 0 and 1 for classification purposes.

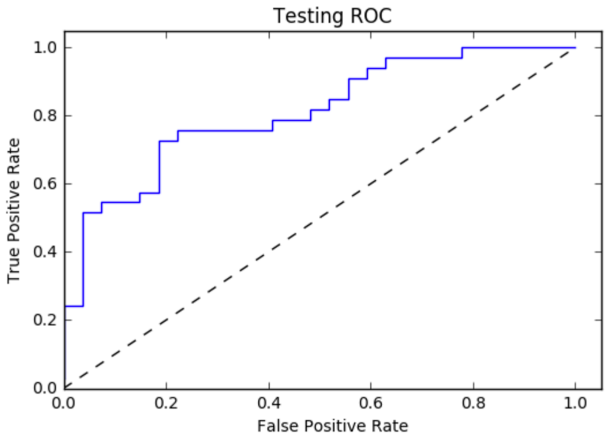
The LSTM for Keras requires a 3-dimensional input tensor with dimensions (*samples* x *timesteps* x *features*). I encoded each integer alarm into a one-hot vector format, with each vector having dimensions (*distinct_ alarms* x *1*). As a result, the matrix for each patient had dimensions (*timesteps* x *distinct_ alarms*) where *timesteps* is the number of alarms available for a given patient. Again, this is one advantage of using an LSTM network over other sequence classification models – variable length sequences pose no problem to the model training. In order to account for this, sequences were padded to the maximum sequence length, after which a masking layer was applied to hide the timesteps that had been added artificially.

V. Results

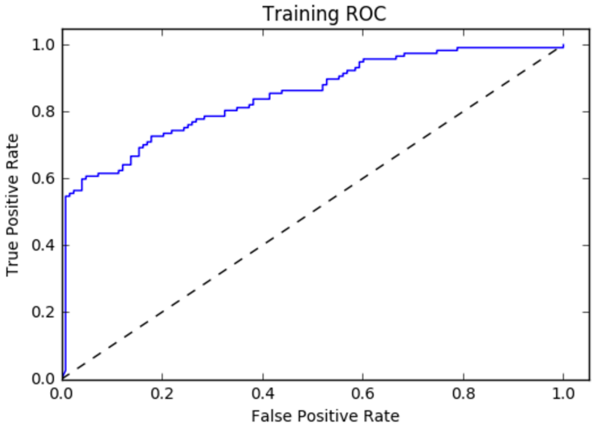
The LSTM model was trained by first setting aside 20% from the dataset for model evaluation purposes. The model was then trained for 300 epochs on the rest of the data (about 240 samples). This process was repeated twice, each time with a randomly selected subset of samples to train on. A third attempt was made, training the model for 1000 epochs on a different training/test set.

Trial 1





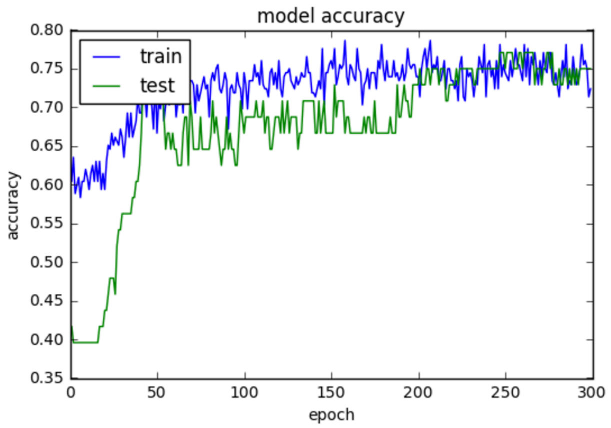
```
AUC: 0.811448
```

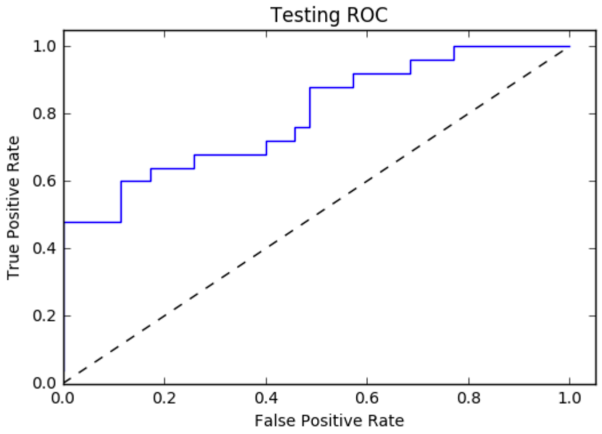


```
AUC: 0.846675
```

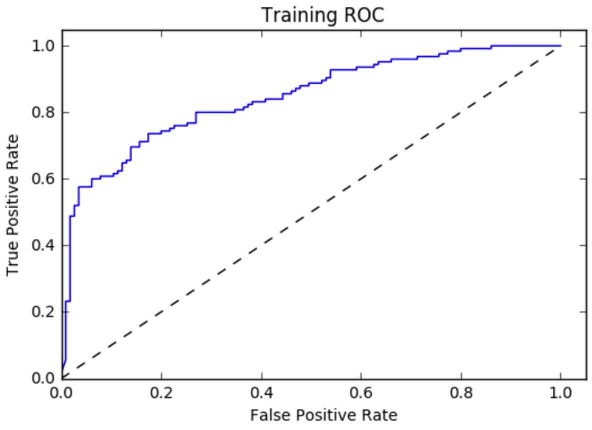
The first model training attempt yielded an accuracy as high as 80% and a training/testing AUCs of 0.85/0.81.

Trial 2





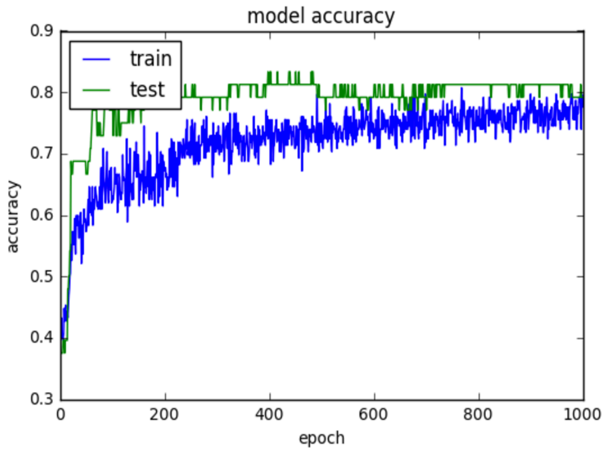
```
AUC: 0.795429
```

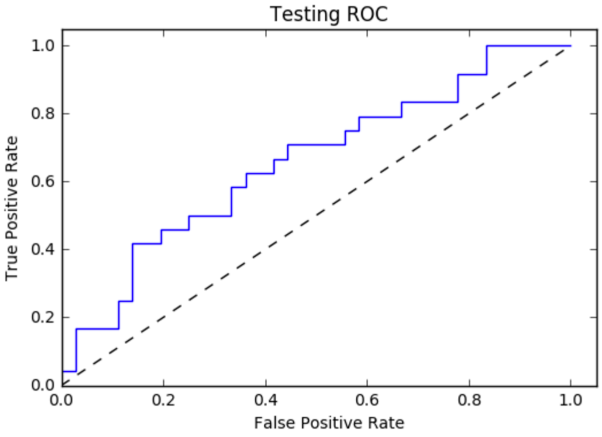


```
AUC: 0.845183
```

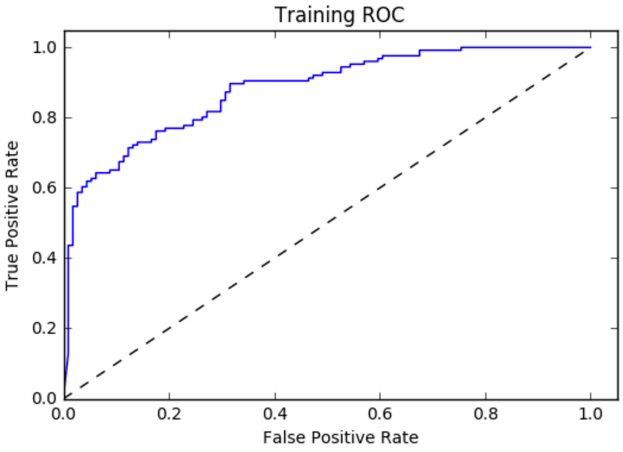
The second model training attempt yielded an accuracy as high as 79% and a training/testing AUCs of 0.85/0.79.

Trial 3





```
AUC: 0.657407
```



```
AUC: 0.880082
```

The final training attempt yielded a much higher training accuracy and AUC at the cost of a worse testing AUC, likely due to overfitting.

VI. Discussion

It appears that first attempts at training an LSTM network have been reasonably effective at classifying sequences as code blue or control, reaching AUCs as high as 0.81. For comparison, one can examine the results from a paper written in 2016 by Georgia Tech researcher Edward Choi [1]. Choi et al. trained RNNs for early detection of heart failure using electronic health record data, obtaining an AUC of 0.7768. Though the paper's timescale is at a larger granularity (up to 12 months as opposed to a few days) and they used a simplified model (a Gated Recurrent Unit instead of a LSTM), their overarching goal is similar enough to use the paper as a benchmark. To that extent, we happily report that the LSTM model's performance is comparable. Before the model can be fully evaluated, hyperparameters (i.e. number of hidden units, dropout layers, etc.) must be tuned completely and the network must be trained on more data.

The work completed so far serves as proof-of-concept. We reason that LSTMs, with proper data augmentation, training, and evaluation, are likely to out-perform existing sequence classification algorithms for sequences of hospital alarms. However, we propose several modifications to the current implementation in order to see the maximum gain in performance:

1. Implement random sampling of patient data to increase the number of training samples, especially for code blue patients.
2. Implement an "online" algorithm that uses pre-trained LSTM models at n-hour intervals for forecasting code blue events before they happen.
3. Implement metrics that effectively capture the model's performance:
    a. Sensitivity of lead time; percentage of code blue patients correctly forecasted.
    b. Alarm frequency reduction rate; "1 – ratio of hourly rate of positive predictions from a trained sequence classifier to the hourly rate of monitor alarms based on the data from control patients." [7]
    c. Work-up to detection ratio; how many false positives can be introduced from the classifier when a single true positive is achieved [7].
4. Employ cross-validation to tune model parameters and training.

VII. Conclusion

With more data and more effective evaluation procedures, we hope to improve the LSTM's performance and develop a state-of-the-art sequence classifier that could be potentially be put into practice in the clinic. Such improvements could lead to decreased volume of alarms in an ICU environment as well as a warning system for patients who are likely to experience cardiac arrest.

VIII. Acknowledgment

I would like to thank Prof. Xiao Hu and the Hu lab for letting me collaborate on this project as part of my studies at U.C. Berkeley.

IX. Works Cited

1. Choi, Edward et al. (2016) Using recurrent neural network models for early detection of heart failure onset. *Journal of the American Medical Informatics Association.*
2. Drew BJ, Harris P, Zègre-Hemsey JK, Mammone T, Schindler D, Salas-Boni R, et al. (2014) Insights into the Problem of Alarm Fatigue with Physiologic Monitor Devices: A Comprehensive Observational Study of Consecutive Intensive Care Unit Patients. *PLoS ONE* 9(10): e110274. https://doi.org/10.1371/journal.pone.0110274
3. Bai, Yong et al. (2015) Integrating monitor alarms with laboratory test results to enhance patient deterioration prediction. *Journal of Biomedical Informatics.*
4. Salas-Boni, R., Bai, Y., & Hu, X. (2015). Cumulative Time Series Representation for Code Blue prediction in the Intensive Care Unit. *AMIA Summits on Translational Science Proceedings*, *2015*, 162–167.
5. Karpathy, A. (2015, May 21). The Unreasonable Effectiveness of Recurrent Neural Networks. Retrieved May 5, 2017, from http://karpathy.github.io/2015/05/21/rnn-effectiveness/
6. Olah, C. (2015, August 17). Understanding LSTM Networks. Retrieved May 5, 2017, from http://colah.github.io/posts/2015-08-Understanding-LSTMs/.
7. Bai, Yong et al. (2016). Is the Sequence of SuperAlarm Triggers more Predictive than Sequence of the Currently Utilized Patient Monitor Alarms? *IEEE Transactions on Biomedical Engineering.*