LSTM Modeling of Perinatal Fetal Heart Rate

Philip A Warrick[1] and Emily F Hamilton[1,2]

[1] PeriGen, Inc, Montreal, Canada
[2] Department of Obstetrics and Gynecology, McGill University, Montreal, Canada

Abstract

Objective: We use Long Short Term Memory (LSTM) units to model perinatal fetal heart rate. We assess the effect of modeling variants such as dropout, multiple cells/layer, multiple layers on model precision and parameter count. Motivation: Linear models can model short-term signal characteristics. An LSTM should do this as well but in addition, retain important longer term characteristics and do so with a low parameter count model. Methods: Recordings of 115 normal, 115 metabolic acidotic and 44 severely pathological (P) fetuses were detrended and downsampled before LSTM modelling. Results: A single-unit LSTM model produced a succinct (4 state parameters), and expressive model (mean variance accounted for of 87.7%) without overfitting. Larger numbers of units N gave marginal improvement at the cost of much higher parameter count.

1. Introduction

Labour and delivery is routinely monitored electronically with sensors that measure maternal uterine pressure (UP) and fetal heart rate (FHR), a procedure referred to as cardiotocography (CTG). Temporary decreases in FHR are known as decelerations and reflect events such as compression of the umbilical cord by uterine contractions, malfunction of the fetal heart muscle, or premature separation of the placenta. Generally, larger insults are indicated by recurring episodes of deep, long decelerations whose onsets occur late with respect to the uterine contractions.

Characterizing the fetal heart rate (FHR) through modelling can give insights into the fetal cardiac state and overall fetal well-being. While linear approaches such as autoregressive models have been used to model short-term cardiac signal characteristics [1, 2], in this study we modelled perinatal FHR using a Long Short Term Memory (LSTM) unit. The LSTM architecture should match this linear performance, but in addition, it should retain important longer term characteristics due to its internal nonlinear and recurrent connections. We also hypothesize that it should do so with low parameterization. In particular, we focus on small architectures, as small as a single LSTM unit, to assess their expressive ability. We assess the effect of modeling variants of dropout, multiple cells/layer and multiple layers on the modelling accuracy.

2. Data

We used CTGs from singleton, term pregnancies having no known congenital malformations, with at least 90 min of tracing just prior to delivery. 115 of the cases were normal with umbilical cord base deficit (BD)< 8, 115 had developed metabolic acidosis (BD> 12 mmol/L, with no apparent neurological injury, 5^{th} percentile of fetuses) and 44 were severely pathological fetuses (confirmed signs of neurological injury, $< 1^{st}$ percentile of fetuses). The data come from hospitals that did routine umbilical cord blood gas measurements shortly after birth.

3. Methods

3.1. Overall processing

After cleaning and downsampling the FHR signal in a preprocessing step, several LSTM model variants were used to predict the next sample in the time series.

3.2. Preprocessing

The CTG data was recorded at 4 Hz in a clinical setting, so it was subject to specific types of noise. The loss of sensor contact can temporarily interrupt the UP or FHR signals, and fetal movement or interference from the (much lower) maternal heart rate can corrupt the FHR. These event caused the signal to drop sharply to much lower value followed by a sharp signal restoration. We first detected and bridged interruptions. Then the data was detrended with high-pass filter selected to pass a long contraction or deceleration (f_c=$\frac{1}{220s} = 4.5 \times 10^{-3}$ Hz). Finally the signal was downsampled to 0.25Hz to focus on the lower frequency band of accelerations and decelerations.

 ISSN: 2325-887X DOI: 10.22489/CinC.2019.419

3.3. LSTM description

LSTM units are recurrent neural networks with feedback connections that allow recent events to be stored in the form of internal activations. Back-propagation through time (BPTT - [3]) and Real-time recurrent learning (RTRL - [4]) have been the conventional algorithms for learning what to put into the short-term memory, but they require long learning times or do not work at all [5]. In addition they fail to bridge gaps in the more distant past (ie. greater than 10 steps) due to back-propagated error signals that either vanish (causing long learning times) or explode (causing oscillating weights).

LSTM overcomes error back-propagation problems by using a gradient based algorithm (using elements from both BPTT and RTRL) whose error flow through its internal states is forced to be constant (rather than exploding or vanishing). The basic LSTM unit is a memory block containing one or more memory cells and three multiplicative and adaptive gating units shared by all cells in the block. These input, forget and output gates learn to control what input information to store in the memory, how long to store it and when to release it to the output, respectively. The internal memory is provided by a recurrently self-connected linear unit that can recirculate activation and error signals indefinitely, providing short term memory storage for extended periods of time.

An LSTM unit has 16 learned weights, but an adaptive signal model can be described by the 4 parameters of its internal states (i.e., input, forget and 2 outputs).

3.4. LSTM networks

LSTM networks typically operate over a time dimension $t \in \{1, 2, \cdots, M\}$ and two spatial dimensions L layers with N_i units per layer, where $i \in \{1, 2, \cdots, L\}$. The LSTM layers are formed by multiple units $j \in \{1, 2, \cdots, N_i\}$.

Figure 1 shows an LSTM unit that has been "unrolled" over multiple time steps $t-1$, t and $t+1$. This unrolling refers to the inherent recurrent nature of the LSTM where the hidden state $\boldsymbol{h}_{t-1}$ and input $\boldsymbol{x}_t$ form a composite input to the unit at time t. LSTM units effectively share the same weights over each unrolled timestep.

At time step t, each unit ij accepts a vector of inputs $\boldsymbol{x}_t^i$ from the previous layer and a vector of hidden states $\boldsymbol{h}_{t-1}^i = h_{t-1}^{ik}, k \in \{1, 2, \cdots, N_i\}$ from the same layer and generates the hidden state h_t^{ij}. LSTM units in the same layer therefore share the same composite inputs but each unit generally has its own set of weights. In Figure 1 multiple units of the same layer are indicated along the diagonal arrows.

The first layer input $\boldsymbol{x}_t^1$ is simply the raw input data, in our case the scalar FHR signal. Multiple layers are constructed by using the hidden states $\boldsymbol{h}_{t-1}^{i-1} = h^{i-1,k}$, $k \in \{1, 2, \cdots, N_{i-1}\}$ of previous layer $i-1$ as layer i input $\boldsymbol{x}_t^i$, as shown in Figure 1 where the layers are shown vertically.

3.5. LSTM FHR models

Using the LSTM as a building block, we formed networks with varying numbers of units per layer as well as the number of layers.

We used Keras and Tensorflow implementations of LSTM because they allowed us to use an graphics processing unit (GPU), the NVidia Tesla TitanX, to efficiently perform parallel sequence training. We used a 12 GB GPU, Tensorflow, Keras and the keras all-CUDA implementation of LSTM, keras.layers.CuDNNLSTM, for performance. Fold training times typically were on the order of 12 hours/fold using this configuration.

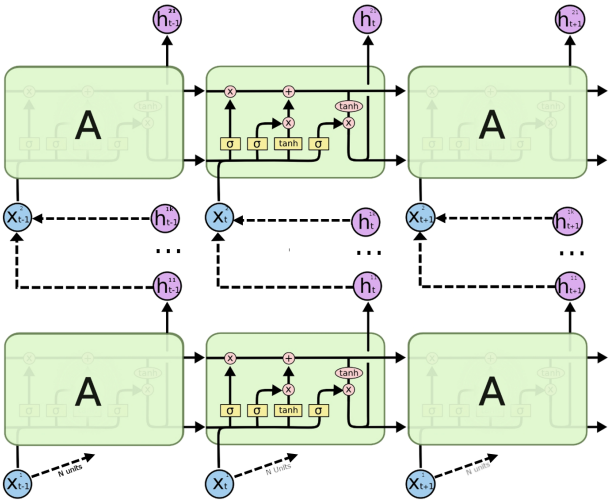

Figure 1. LSTM network dimensions. The unrolled time dimension is shown horizontally with identical unit structures 'A' at each time step. The diagonal arrow indicates the direction where N_i units connect to a common input while multiple layers are formed along the vertical dimension. Adapted from [6].

3.6. Regression training and evaluation

To prepare the data for training, we normalized the data by mapping the overall minimum and maximum values to -1 and +1, respectively. Then we split the signals into lengths of M=4500 samples for batch training. The targets were aligned with the input such that at time t, we predicted FHR($t+1$). We then performed ten-fold CV using data proportions of 80% training, 10% validation, 10% test for each fold. As a loss function we used the mean squared error (MSE) on valid signal and employed early

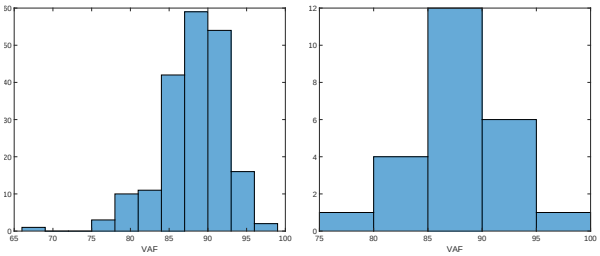

Figure 3. Histograms of typical recording VAFs in a single fold for training (left, $n = 198$) and testing sets (right, $n = 24$).

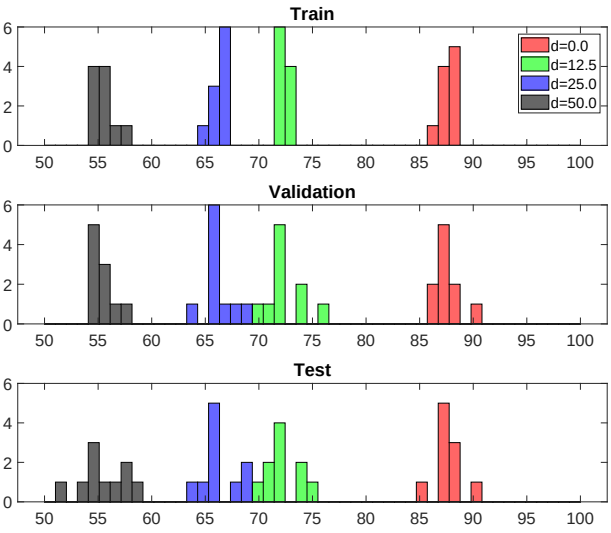

Figure 4. Histograms showing effect of dropout percentage $d \in \{0, 12.5, 25, 50\}$ on mean fold VAFs for a single LSTM unit.

stopping observing the validation loss over 20 epochs. After training, we concatenated predicted signals from the same recording and calculated the variance accounted for VAF = 100 $\times(1 - \sigma_e^2/\sigma_o^2)$, where σ_o and σ_e are the true and prediction error FHR signal energies, respectively.

We used a two-sided t-test to compare the fold VAFs of experiment pairs for the null hypothesis that their means were equal.

4. Results

Fig. 2 shows typical measured and predicted FHR for a signal excerpt of approximately 60 minutes duration showing their similar signal characteristics (the VAF was 80%).

Fig. 3 shows typical test VAFs for a single fold for training and testing sets. The mean and variance is similar for both, indicating that there was little overfitting.

Fig. 4 shows the effect of dropout on the mean fold VAFs for a single LSTM unit. It is clear that as the amount of dropout increases, the fidelity of the model decreases (all comparisons $p < 0.0001$), approximating the effect of increasingly noisy signal.

Fig. 5 shows the effect of the number of LSTM units for a single layer on the mean fold VAF. The VAF distribution with 100 units (mean 91.2%) was very similar to 10 units (mean 91.8%, $p = 0.079$), but a single unit was only slightly inferior (mean 87.7%, $p < 0.0001$). Again the training, validation and test distributions were very similar for the same number of units, indicating that there was little overfitting. For reference, a baseline predictor that predicts using the last measured sample, i.e., predicted FHR($t + 1$) = measured FHR(t) is also shown. The superiority of all LSTM networks over the baseline (mean 73.8%) is apparent ($p < 0.0001$ for comparison with N=1).

Fig. 6 shows the effect of the number of layers with a single unit per layer on the mean fold VAF. A two-layer network performed slightly better (mean 89.2%) than a single layer network (87.7%, $p = 0.0022$).

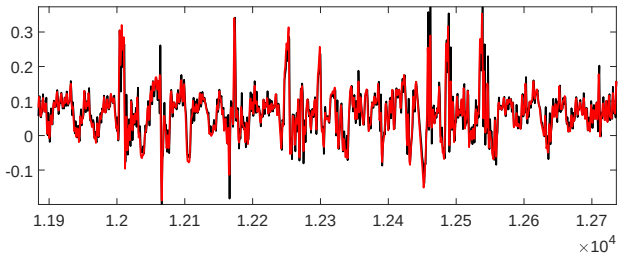

Figure 2. Typical measured and predicted FHR (∼60 min excerpt) with a VAF of 80%.

5. Conclusions

The model hyperparameter search space was not exhaustive in this study, but the tendencies are clear. A single-unit LSTM model produced a succinct (4 state parameters) and expressive (mean VAF$\sim=87\%$) signal predictor model without overfitting. Larger numbers of units N gave marginal improvement at the cost of a much higher parameter count. Increasing dropout degraded prediction, simulating the effect of worsening signal quality. We will perform a finer hyperparameter search in future work to tune this model further.

6. Disclosure

This research was funded by PeriGen Inc.

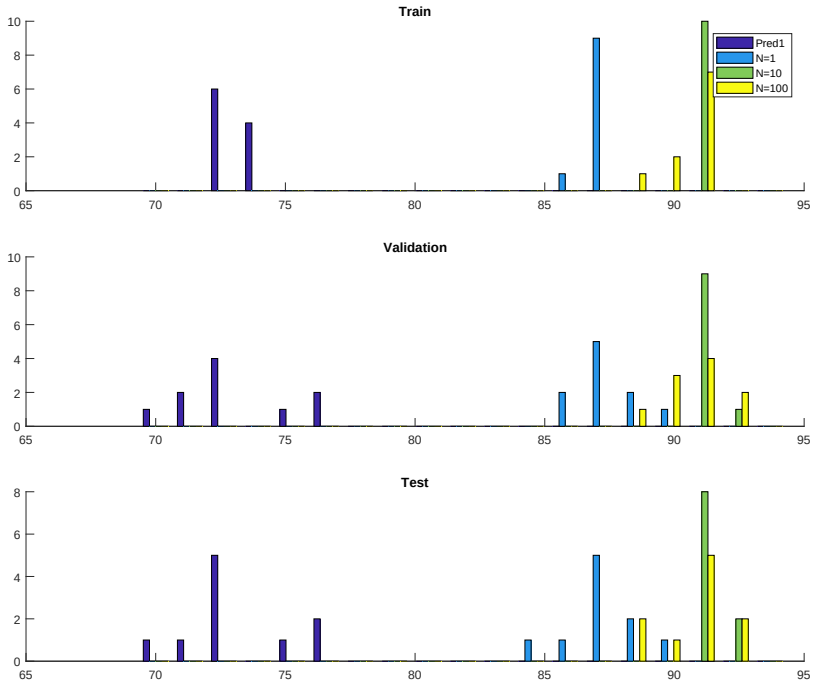

Figure 5. Histograms showing effect of number of units $N \in \{1, 10, 100\}$ on mean fold VAFs for a single LSTM layer. The baseline last-sample predictor 'Pred1' is also shown.

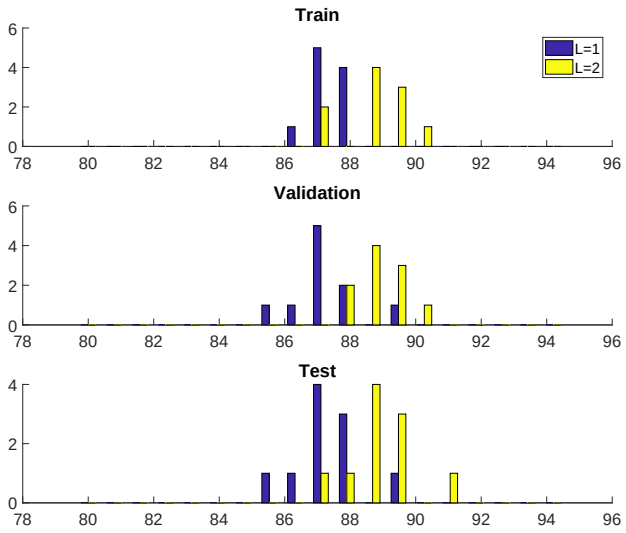

Figure 6. Histograms showing effect of number of layers $L \in \{1, 2\}$ on mean fold VAFs with a single LSTM unit per layer.

References

[1] Cerutti S, Civardi S, Bianchi A, Signorini M, Ferrazzi E, Pardi G. Spectral analysis of antepartum heart rate variability. Clin Phys Physiol Meas 1989;10:27–31.

[2] Signorini M, Magenes G, Cerutti S, Arduini D. Linear and nonlinear parameters for the analysis of fetal heart rate signal from cardiotocographic recordings. IEEE Transactions on Biomedical Engineering 2003;50(3):365–374. ISSN 0018-9294.

[3] Werbos P. Generalization of backpropagation with application to a recurrent gas market mode. Neural Networks 1988; 1(4):234–242.

[4] Robinson AJ, Fallside. F. The utility driven dynamic error propagation network. Technical Report Technical Report CUED/F-INFENG/TR.1, Cambridge University Engineering Department, 1987.

[5] Gers FA, Schmidhuber J, Cummins F. Learning to forget: Continual prediction with LSTM. Neural Computation 2000.;12(10):2451–2471.

[6] Olah C. Understanding lstm networks, 2015. URL `https://colah.github.io/posts/2015-08-Understanding-LSTMs/`.

Address for correspondence:

Philip A. Warrick
PeriGen Inc. (Canada)
245 Victoria Avenue, suite 600
Montreal, Quebec H3Z 3M6 Canada
philip.warrick@perigen.com