Antenatal Fetal Heart Rate Acceleration Detection

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

During the antenatal period non-stress test, fetal heart rate (FHR) is routinely measured non-invasively by cardiotocography. The purpose of this study was to measure the performance of FHR acceleration detection for fetuses having a gestational age (GA) 28-40 wks, comparing our existing detector for term fetuses to one based on a hidden semi-Markov model (HSMM) and another based on an LSTM (Long short-term memory) recurrent neural network.

1. Introduction

Labour and delivery is monitored electronically with sensors that measure and record maternal uterine pressure (UP) and fetal heart rate (FHR), a procedure referred to as cardiotocography (CTG). The objective of this monitoring is to detect the fetus at substantial risk of hypoxic injury so that intervention can prevent its occurrence. However, in the antenatal period long before delivery, fetal heart rate (FHR) is also routinely measured during the non-stress test. Clinicians view the presence of FHR acceleration events during this test as a positive indicator of fetal health. Fetal heart rate (FHR) accelerations (ACC) are elevations in the heart rate over the baseline, having a minimum amplitude of 15 beats per minute, lasting 15 s to several minutes.

While we have developed a detector of FHR events for term labour, we have not applied it in the smaller amplitude, lower signal-to-noise (SNR) antenatal context. The purpose of this study was to measure the performance of FHR acceleration detection for fetuses having a gestational age (GA) 28-40 wks, comparing our reference detector to one based on a hidden semi-Markov model (HSMM) and another based on an LSTM (Long short-term memory) recurrent neural network.

2. Data

From a dataset of 72 antenatal CTG recordings for gestational age (GA) 28-40 wks, with durations ranging from 20 min to several hours, an expert obstetrician (one of the co-authors) manually annotated the ACCs. Each antenatal GA group included at least 5 recordings.

3. Methods

3.1. 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 interference from the (much lower) maternal heart rate can corrupt the FHR. These both appeared in the signal as a sharp drop to much lower amplitude followed by a sharp signal restoration. As described in [1], we preprocessed the data to bridge interruptions with linear interpolation.

For HSMM preprocessing, we detrended the signals by a moving window median filter of 20-min past samples. For LSTM the detrending was performed with a high-pass filter selected to pass a long contraction or deceleration without incurring excessive filter delay. We chose a filter with a cutoff frequency of $\frac{1}{220s} = 4.5 \times 10^{-3}$ Hz as a compromise between these competing demands.

The resulting signals where then decimated to 0.25 Hz to reduce feature and training computational requirements.

3.2. Reference detector

The reference detector for this study is an FDA-approved detector of FHR events for term labour (PeriCALM Patterns^{TM}). It employs a combination of ad-hoc rules, signal processing and feed-forward neural-network classification to estimate FHR baseline and variability, and to detect acceleration and deceleration patterns.[2]

3.3. HSMM detector

We used an HSMM adapted from the Physionet Logistic Regression HSMM [3] because it accounts for the state durations of the ACCs. We allowed two states (ACC and non-ACC) and used a fixed Gaussian durational model with forced transitions only after the ACC state.

Computing in Cardiology 2016; VOL 43 ISSN: 2325-887X DOI:10.22489/CinC.2016.259-501

3.4. LSTM detector

Whereas the reference detector employed feed-forward neural networks, the LSTM networks of this study are recurrent, and their feedback connections allow recent events to be stored in the form of internal activations. Backpropagation through time (BPTT - [4]) and Real-time recurrent learning (RTRL - [5]) 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 [6]. 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, respectively, what input information to store in the memory, how long to store it and when to release it to the output. 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.

We used the CUda RecurREnt Neural Network Toolkit (CURRENNT) [7] because it allowed us to use an graphics processing unit (GPU), the NVidia Tesla C2050, to efficiently perform parallel sequence training with network architectures significantly larger than those used in the reference detector.

3.5. Feature calculation

For the LSTM detector, we performed Karhunen-Loève (KL) decomposition, retaining 8 components corresponding to the highest eigenvalues. The eigenanalysis used observation vectors from 40 s sliding windows. We also added a component corresponding to the reconstruction energy of the KL components, as well as the median signal, the detrended high pass and its complement low-pass signal. For the HSMM detector, features were derived from a discrete cosine transform (DCT) decomposition with a 160 s window and 3 components of a 12-level decomposition (dct6, dct10, dct12) as well as a median signal and a total energy component. Adding more components resulted in poor training performance. The DCT decomposition used with the HSMM detector can be considered an fixed-basis approximation to KL decomposition [8].

3.6. Training

We used 10-fold cross validation for training. Each sample was labelled based on its overlap with the expert marking. Samples occurring in low-quality signal intervals were labelled as noise, resulting in a three-class labelling (noise, non-accel and accel). For the LSTM case, the cross-validation included train, validation and test partitions. The eigenanalysis was performed per-fold on the training set and applied to the validation and test sets. Similarly, per-fold normalization for zero-mean and unit standard deviation was computed for the training data and then applied to the other sets. Normalization is critical for LSTM numerical stability. To overcome gradient-based training susceptibility to local minima, we trained multiple (5) neural nets per fold with randomized initial weights.

Batch training was done on the approximately 56 training sequences with 10 parallel sequences per "mini-batch". The LSTM network architecture included three hidden layers of 50, 100 and 34 cells, respectively. These values were chosen by experiment to be as large as possible while remaining within GPU memory availability and reasonable cross-validation duration (e.g., 12 hours). The 12 features were applied to the input layer and the output layer used a softmax activation function followed by a cross-entropy objective function for the classification. To provide better contextual information, the training included a 2 minute "lookahead", such that predicted output was allowed to consider features from up to 2 minute future features. This lookahead value was a compromise between real-time delay and potential improved accuracy. For better generalization, zero-mean gaussian noise of standard deviation 0.6 was added to each input, as suggested in [7]. Other parameters were set to the CURRENNT default values. Training continued for each batch epoch until the overall objective error function did not improve on the validation set after 20 epochs (i.e., early stopping). The neural network with the best performance on the validation set was retained for use with the test set.

3.7. Evaluation

Evaluation was based on performance on the test set, whose independence was ensured by restricting all tuning parameters (eigenanalysis, normalization, network architecture and weights, early stopping, etc.) to observation of training and validation sets alone, as described above. An ACC annotated by the expert was deemed detected by one of the detectors if any predicted ACC sample overlapped with the expert event. We report overall and per GA-group sensitivity (Sens) and positive predictive value (PPV) based on this detection definition.

	Sens(%)	PPV(%)	Sens+PPV(%)
Train	79.6±5.6	79.3±4.6	158.9±3.5
Validation	79.4±4.7	79.4±7.4	158.9±9.4
Test	74.8±10.2	81.4±9.3	156.2±14.0

Table 1. For each data partition over all GA groups, mean and standard deviation over all 10 folds for sensitivity (Sens), positive predictive values (PPV) and Sens+PPV for the LSTM acceleration detector.

4. Results

The overall LSTM results are shown in Table 1. The training results were modestly better than the validation and test results with Sens and PPV both at close to 80% and a bit less for test Sens (74.8 %), suggesting that early stopping effectively avoided overfitting during training.

The per GA-group performance is shown in Fig. 1 for all three detectors, as well as the number of ACC events marked by the expert for each group. The reference and LSTM detectors had comparable performance with an average of Sens and PPV of approximately 80% over all GA groups. Both were consistently superior compared to HSMM in PPV while HSMM sensitivity was more stable across the GA groups with slightly higher Sens at low GA. PPV tended to degrade with earlier GA for all detectors.

5. Conclusions

Both the LSTM and HSMM detectors have signal processing and machine learning advantages over the reference detector. Information capture should be more complete since features are based on orthogonal signal decompositions rather than manually selected features having direct correspondence with familiar clinical measures. Furthermore, training had a greater learning capacity (e.g., 100k LSTM cells compared to tens of feed-forward neural network neurons) with a much smaller set of tuning parameters and as such the comparable results between LSTM and the reference reflect a promising first iteration at what is possible with more modern computational hardware and with far less development effort than required for the reference detector (multiple human-years).

All detectors degraded in the lower SNR conditions of earlier GA. In the next phase of this study, models that account for GA, and based on a larger dataset, should improve performance. We also intend to investigate further which features are most discriminating and use a common set to better compare both detectors. Finally, despite the reduced algorithm complexity of both the LSTM and HSMMM detectors further searching of the tuning parameter space is required to achieve better performance.

6. Disclosure

This research was funded by PeriGen Inc.

References

[1] Warrick PA, Hamilton EF, Precup D, Kearney RE. Identification of the dynamic relationship between intra-partum uterine pressure and fetal heart rate for normal and hypoxic fetuses. IEEE Transactions on Biomedical Engineering June 2009;56(6):1587–1597.

[2] Warrick P, Hamilton E, Macieszczak M. Neural network based detection of fetal heart rate patterns. In Neural Networks, 2005. Proceedings. 2005 IEEE International Joint Conference on, volume 4. 2005; 2400–2405.

[3] Springer DB, Tarassenko L, Clifford GD. Logistic Regression-HSMM-Based Heart Sound Segmentation. IEEE Transactions on Biomedical Engineering April 2016; 63(4):822–832. ISSN 0018-9294.

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

[5] 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.

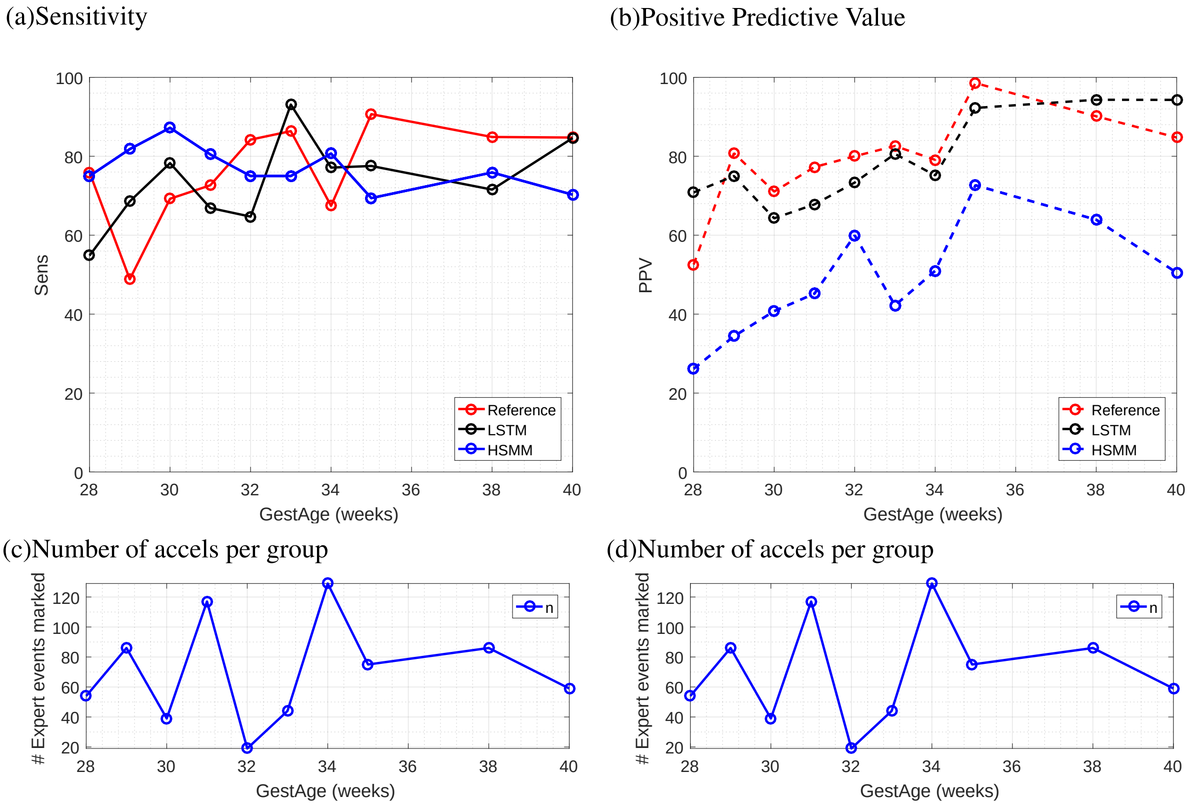
[6] Gers FA, Schmidhuber J, Cummins F. Learning to forget: Continual prediction with LSTM. Neural Computation 2000.;12(10):2451–2471.

[7] Weninger F. Introducing CURRENNT: The Munich Open-Source CUDA RecurREnt Neural Network Toolkit. Journal of Machine Learning Research 2015;16:547–551. URL http://jmlr.org/papers/v16/weninger15a.html.

[8] Haykin S. Adaptive Filtering. Fourth edition. Upper Saddle River, New Jersey: Prentice-Hall, 2002.

Address for correspondence:

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


(a)Sensitivity

(b)Positive Predictive Value

(c)Number of accels per group

(d)Number of accels per group

Figure 1. (a) Sensitivity and (b) PPV for the reference (red), LSTM (black), and HSMM (blue) detectors. (c,d) Number of accels per group.