

Intrapartum Fetal-State Classification using Long Short-Term Memory Neural Networks

Philip A. Warrick¹ and Emily F. Hamilton^{1,2}

¹ PeriGen, Inc, Montreal, Canada

² Department of Obstetrics and Gynecology, McGill University, Montreal, Canada

Abstract

We classified perinatal cardiotocography (CTG) records according to their outcome severity using long short-term memory (LSTM) recurrent neural networks. Our classifier classified pathological with 44.7% sensitivity (Se) and 55.3% positive predictive value (PPV), while normal cases had 40.0% Se and 46.7% PPV. The more difficult intermediate MA cases had 33.5% Se and 29.3% PPV.

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.

We wanted to classify cardiotocography (CTG) records according to their outcome severity. In this study, our features included the orthogonal components from Karhunen-Loève (KL) decomposition, an indication of UP-FHR interaction via their mutual information, several baseline trends, and an indication of signal quality. As a classifier we used several layers of long short-term memory (LSTM) recurrent neural networks.

Other works have used attempted to classify perinatal CTG records using feedforward neural networks [1,2], and we have used Support Vector Machines [3] for this task. These approaches are generally static in that they tend to consider one sliding window at a time. We hypothesized that the ability of LSTMs to include recent and distant history could be beneficial to this problem.

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. 70 the cases were normal (N), 71 had developed MA (umbilical cord base deficit \leq

12 mmol/L) and 47 severely pathological (P) fetuses. The data come from hospitals that did routine umbilical cord blood gas measurements.

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 [4], we preprocessed the data to bridge interruptions with linear interpolation. The resulting signals were then decimated to 0.25 Hz to reduce feature and training computational requirements.

3.2. Feature calculation

Features were derived from a median-detrended FHR signal decomposed by the Karhunen-Loève (KL) transform, the mutual information (MI) of the UP-FHR signal pair and an indication of signal quality where the preprocessing had bridged the data. We also the median signal, the detrended high pass and its complement low-pass signal.

For the KL decomposition, we retained 8 components corresponding to the highest eigenvalues. The associated eigenanalysis used observation vectors from 40 s sliding windows.

We calculated MI for both the preprocessed UP and FHR signal pair $u(t)$ and $f(t)$ as well as their instantaneous phases $\phi_u(t)$ and $\phi_f(t)$ as described in [5]. Binning was used for the probability densities $p(u)$, $p(f)$ and $p(u, f)$ required for the MI calculation. Surrogate UP-FHR pairs generated by the amplitude-adjusted Fourier Transform technique were used to test MI significance.

Class	n	$n(\%)$	Se(%)	PPV(%)
N	70	37.2	40.0	46.7
MA	71	37.8	33.8	29.3
P	47	25.0	44.7	55.3
Total	188	100.0		

Table 1. Number, proportion and classification results by class

3.3. LSTM classifier

The LSTM networks of this study are recurrent, and their feedback connections allow recent events to be stored in the form of internal activations. Back-propagation through time (BPTT - [6]) and Real-time recurrent learning (RTRL - [7]) 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 [8]. 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.

The classifier was implemented in Python using the Keras library with a TensorFlow backend, which provides efficient functionality on CPUs and GPUs.

3.4. Training

The LSTM neural network was trained using the features as inputs and the outcome class (N, MA or P) as targets at each timestep. The final prediction generated by the trained neural network classified each recording. Ten-fold cross-validation used training, validation and testing fold partitions.

Efficient batch-oriented training requires fixed-length input; therefore we defined a maximum length, max_length , equal to 5 hrs, for the CTG signals in the dataset. Each CTG signal that had a length less than max_length was end-padded with zeros. A zero label was also added to the sequence of labels to be referred as PAD class. Sam-

ples of this class did not contribute to the training loss. On the other hand, those CTG signals that are longer than max_length were split into multiple sequences in such that each new sequence length was less than or equal to max_length .

We used 10-fold cross validation for training which 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. To overcome gradient-based training susceptibility to local minima, we trained multiple (2) neural nets per fold with randomized initial weights.

Batch training was done on the approximately 150 training sequences with 80 parallel sequences per “mini-batch”. The LSTM network architecture included three hidden layers of 75, 150, and 50 cells, respectively. 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. Other parameters were set to the Keras 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.

3.5. 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 CTG was deemed correctly classified if its final prediction match the outcome target and we used this to measured sensitivity (Sens, also known as ‘recall’) and positive predictive value (PPV, also known as ‘precision’).

4. Results

Table 1 shows test classification results over all records. P cases were best classified with 44.7% sensitivity (Se) and 55.3% positive predictive value (PPV), while N cases had 40.0% Se and 46.7% PPV. The more difficult intermediate MA cases had 33.5% Se and 29.3% PPV.

5. Conclusions

Predicting fetal outcome is a difficult problem and we have made several simplifying assumptions. Most importantly, we have not considered that the fetal state changes with time: using outcome data sampled just after birth

gives reasonable targets for late labour, but applying these same targets throughout labour likely confused the learning process. We will adopt strategies to account for this in future work.

6. Disclosure

This research was funded by PeriGen Inc.

References

- [1] Tsoulos I, Georgoulas G, Gavrilis D, Stylios C, Bernardes J, Groumpos P. Introducing grammatical evolution in fetal heart rate analysis and classification. In Intelligent Systems, 2006 3rd International IEEE Conference on. 2006; 560–565.
- [2] Jezewski M, Wrobel J, Labaj P, Leski J, Henzel N, Horoba K, Jezewski J. Some practical remarks on neural networks approach to fetal cardiotocograms classification. In Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE. ISSN 1557-170X, 2007; 5170–5173.
- [3] Warrick PA, Hamilton EF, Precup D, Kearney R. Classification of normal and hypoxic fetuses from systems modeling of intrapartum cardiotocography. IEEE Transactions on Biomedical Engineering 2010;57(4):771–779. ISSN 0018-9294.
- [4] 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.
- [5] Warrick PA, Hamilton EF. Mutual information estimates of CTG synchronization. In Computing in Cardiology, volume 42. 2015; 137–139.
- [6] Werbos P. Generalization of backpropagation with application to a recurrent gas market mode. Neural Networks 1988; 1(4):234–242.
- [7] 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.
- [8] Gers FA, Schmidhuber J, Cummins F. Learning to forget: Continual prediction with LSTM. Neural Computation 2000.;12(10):2451–2471.

Address for correspondence:

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

