Demographic Information Initialized Stacked Gated Recurrent Unit for an Early Prediction of Sepsis

Naoki Nonaka[1], Jun Seita[1]

[1] Medical Sciences Innovation Hub Program, Riken, Tokyo, Japan

Abstract

Sepsis is a life-threatening condition that occurs when the body's response to infection causes tissue damage, organ failure, or death. Sepsis is a major public health issue responsible for significant morbidity, mortality, and healthcare expenses. Early detection and antibiotic treatment of sepsis are critical for improving sepsis outcomes, where each hour of delayed treatment has been associated with roughly an 4-8% increase in mortality. Thus, an early detection of sepsis can have significant impact on both patient outcome and reduce in medical expenses.

In recent years, deep neural networks has shown significant improvement in variety of tasks. One of the approach to apply deep neural networks to sequential data is a Recurrent Neural Netowrks (RNNs). In this study, we modify gated recurrent units (GRU), RNNs with gate structure, to predict sepsis from provided Physionet Challenge 2019 dataset. In proposed model, initial value of hidden state in GRU was determined by demographic information of patients, and two-step training was performed with customized loss function.

With the proposed method, we achieved normalized utility score of 0.323 on full test set (Team name: NN-MIH).

1. Introduction

Sepsis is a life-threatening condition that occurs when the body's response to infection causes tissue damage, organ failure, or death. Sepsis is a major public health issue responsible for significant morbidity, mortality, and healthcare expenses. Early detection and antibiotic treatment of sepsis are critical for improving sepsis outcomes, where each hour of delayed treatment has been associated with roughly an 4-8% increase in mortality. Thus, an early detection of sepsis can have significant impact on both patient outcome and reduce in medical expenses.

In recent years, deep neural networks has shown significant improvement in variety of tasks, such as object recognition, machine translation and speech recognition[1][2][3]. Among deep neural networks, Recurrent Neural Networks (RNNs) is one of the approach to deal with sequential data. RNNs can capture underlying structure in sequential data, and have been applied to areas such as speech recognition and text classification[3][4]. Although RNNs can capture time dependencies in sequential data, they suffer from vanishing and exploding gradient problems. To mitigate this problem, RNNs with gate structures such as Long short-term memory (LSTM) or gated recurrent units (GRU) has been proposed[5][6].

In this study, we attempt an early detection of sepsis in the physionet challenge 2019 data [7] using variant of GRU. The main ideas in this research are the following three points.

- GRU initialization with patient demographics
- Loss function based on the utility score
- Two steps training

The rest of the paper is organized as follows: the overview and preprocessing of data is described in Section 2, the model used for early detection of sepsis is described is Section 3, experimental results are shown in Section 4 and study is concluded at Section 5.

2. Preprocess of data

In this section we overview the dataset and describe the pre-processing performed on the challenge data.

The provided data for the challenge is sourced from ICU patients in three separate hospital systems, and data from two hospitals are publicly available. Each sample is recorded every hour and consists of 41 variables with vital signs, laboratory values, demographics and sepsis label. The total number of samples collected from 2 hospitals is 40,336, of which 2,932 is septic. In this study, we attempt to predict sepsis label provided using a total of 40 variables, vital signs, laboratory values and demographics.

Here we describe the pre-processing of data. First, the published data was checked for each variable, and outliers were removed. Specifically, clamping was performed on data having a large deviation from the average value. Subsequently, the logarithm was calculated for following variables: DBP, Resp, PaCO2, AST, BUN, alkalinephos, creatinine, bilirubin direct, glucose, lactate, magnesium, phos-

ISSN: 2325-887X DOI: 10.22489/CinC.2019.153

phate, potassium, bilirubin total, PTT, WBC, fibrinogen and platelets. In addition, for O2sat and SaO2, we calculated the logarithm, after subtracting the maximum value for each variable.

After clamping and logarithm calculation, each variables were standardized by subtracting mean and dividing by variance. Subsequently, missing values were filled with 0 to obtain training data. Simultaneously, we store binary matrix to track whether value was missing or not.

Here after we denote data for patient i as $X^{(i)} = [x_1^{(i)}, x_2^{(i)}, ..., x_{T_i}^{(i)}]$, where T_i is lengths of data for patient i. And $x_t^{(i)} = [x_{v,t}^{(i)}, x_{l,t}^{(i)}, x_{d,t}^{(i)}]$, where $x_{v,t}^{(i)}$, $x_{l,t}^{(i)}$, $x_{d,t}^{(i)}$ is vital sign values, laboratory values and demographics for patient i at time t respectively. $x_{m,t}^{(i)}$ denotes binary matrix that indicates whether there are missing values.

3. Model

In this section, we explain recurrent neural network structure, loss function, sequence alignment in mini batch and over all training strategy.

3.1. Demographic information initialized GRU

One of the approach to deal with sequential data with deep neural networks is Recurrent Neural Networks (RNNs). Nevertheless, vanilla RNNs can suffer from vanishing and exploding gradient problems. To mitigate this problem RNNs with gated structures, such as Gated Recurrent Units (GRU) are used. Although GRU can capture structure in sequential data, it does not have suitable structure to consider static information related to sequential data, such as demographics of patients. Therefore, we propose a demographic information initialized GRU (DIGRU) to consider static information while capturing structure in sequential data. DIGRU determines the initial value h_0 of the hidden state of the GRU using demographics of patients. DIGRU is defined as follows.

$$z_t = \sigma_g(W_z x_t + U_z h_{t-1} + b_z) \tag{1}$$

$$r_t = \sigma_g(W_r x_t + U_r h_{t-1} + b_r) \tag{2}$$

$$h_t = (1-z_t)\otimes h_{t-1} + z_t \otimes \sigma_h(W_h x_t + U(r_t \otimes h_{t-1}) + b_h) \tag{3}$$

$$h_0 = f(x_{d,0}) \tag{4}$$

Where W_z, U_z, W_r, U_r is a weight, b_z and b_r is a bias, h_t is a hidden state at time t, x_t is an input at time t and f is fully connected layer. Hidden state h_t is initialized by demographic information of each patient $x_{d,0}$ and updated by vital sign value $x_{v,t}$ and laboratory value $x_{l,t}$ at each time step.

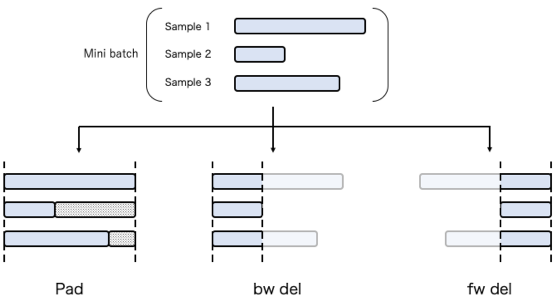

Figure 1. Illustration of sequence alignment in mini batch.

3.2. Loss function

In this competition, the model is evaluated by the utility score that weights the binary classification results. Therefore, the loss function for directly optimizing the utility score was put and the model was trained. The loss function was defined as follows.

$$U = v_t * \log \hat{y} + \left(\frac{u_{n,t} + 2}{w}\right) * \log(1 - \hat{y}) \tag{5}$$

$$v_i = \begin{cases} 1 & (u_{p,t} > 0) \\ 0 & (u_{p,t} \leq 0) \end{cases} \tag{6}$$

Where w is a parameter to control weight for non sepsis prediction, $\hat{y}$ is an output from prediction model. $u_{p,t}, u_{n,t}$ is a utility score given to sepsis prediction ($\hat{y} = 1$) and non sepsis prediction ($\hat{y} = 0$) at time t, respectively.

3.3. Sequence alignment for minibatch

Sequence length of each data differs, however to train neural network, all data in mini batch needs to have same sequence length. To align the sequence length of each sample in mini batch, we used three methods as shown in figure 1.

Pad Sequence length was aligned to longest sample in minibatch. Zero padding was applied to latter part of each sample.

bw del Sequence length was aligned to shortest sample in minibatch. Only first N datapoints were used and latter part was ommited

fw del Sequence length was aligned to shortest sample in minibatch. Only last N datapoints were used.

When samples are aligned by "pad", the entire sequence length is given to the model. On the other hand, in "bw del", only the data at the beginning of ICU entry is given, and in "fw del", data when time passes from ICU entry is given to the model. Especially for data with a long sequence length, data immediately after entering the ICU ("bw del") and late data ("fw del") was given to the model.

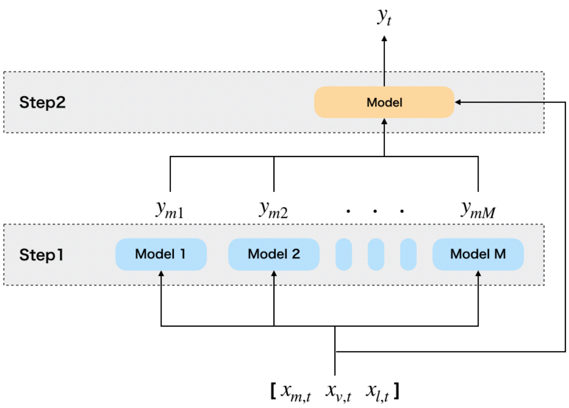

Figure 2. Illustration of 2 step prediction model. Multiple models are trained in step 1. Prediction by each models and input data is combined and given as an input to the model for step 2.

3.4. Stacked model

The model training was divided into two stages as shown in Figure2.

As a first step, training was performed in several different settings. The w parameter, controlling weight for non sepsis prediction, in the loss function was set to a different values and the method for adjusting the length of data in mini batch was changed. Specifically, we trained 6 models, model1 to 4 with “pad” alingment and $w = 40, 20, 50$ and 50 respectively. We used cyclic learning rate scheduler for model4. For model5 and model6 we used “fw del” and “bw del” for sequence alignment with $w = 50$ for the loss function. For all models, layer structure was set to DIGRU, layer normalization [8], DIGRU, layer normalization, fully connected layer, ReLU activation, fully connected layer, as shown in Figure3. We introduce residual connection between each DIGRU layer [9]. As for other hyper parameters, hidden state size was set to 32, AdamW [10] were used for optimization, and learning rate was set to 0.0001. Batch size was set to 512 for model1 to 4 and 4096 for model5 and 6. We set training epochs to 500 and checked utility score of validation dataset every 5 epochs, and chose model with best validation utility score as a final model for each training settings.

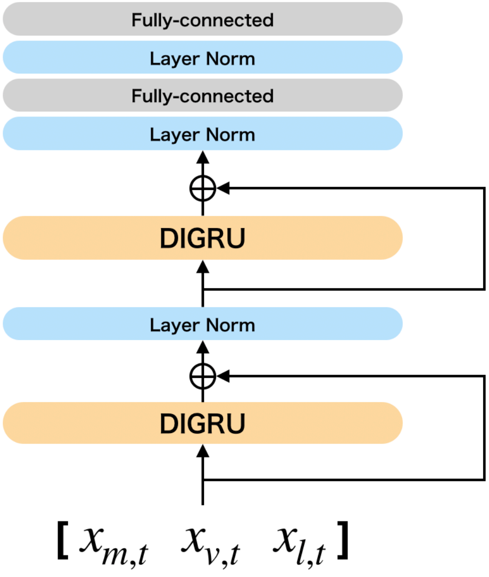

Figure 3. Layer structure of models used in step 1.

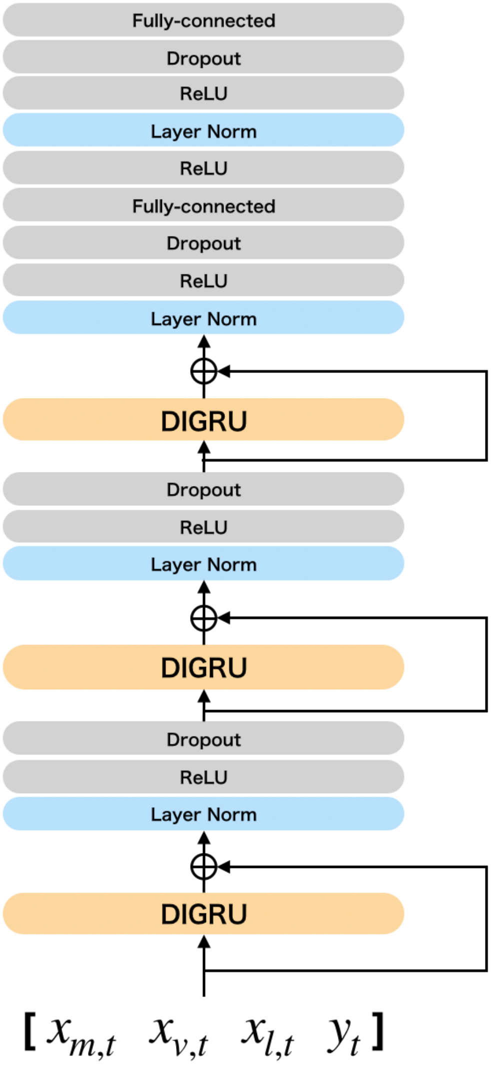

Figure 4. Layer structure of model used in step 2.

As a second step, as an input data, we combine original input variable x_t and $y_t = [y_{1,t}, ..., y_{6,t}]$. Where $y_{M,t}$ is an output from model M given input x_t. As a network structure, we prepared block consists of DIGRU, layer normalization, ReLU and Dropout [11]. We repeated the block for three times and added fully connected layer, ReLU, layer normalization, ReLU, dropout and fully connected layer. We set hidden state size to 32, learning rate to 0.00001, batchsize to 512 and used AdamW as an optimizer. Weight for loss function was set to 50 and mini batch was aligned by "pad".

4. Results and Discussion

A model using DIGRU was applied to predict sepsis. The published data was divided into train/valid/test dataset and the model was trained using train dataset and valid dataset. The utility score in train/valid/test dataset was calculated using the trained model. The results are shown in table1.

Table 1. Utility scores for each model used.

Models	Train	Valid	Test
model1	0.4386	0.4143	0.4096
model2	0.4403	0.4161	0.4055
model3	0.4258	0.4159	0.4214
model4	0.4432	0.4116	0.4301
model5	0.1942	0.1795	0.1893
model6	0.2206	0.1912	0.2001
Final Model	0.4513	0.4228	0.4168

The utility score did not change significantly when changing the parameter w in the loss function. In addition, there was no significant difference in the utility score obtained when the scheduler was set to cyclic. On the other hand, when the sequence alignment in the mini batch was changed to "fw del" or "bw del", the utility score decreased significantly. The prediction results of the model combining the prediction results from model1 to 6 exceeded the individual models in the validation dataset.

With the proposed method, we achieved normalized utility score of 0.323 on full test set (0.414 on test set A, 0.373 on test set B and -0.174 on test set C; Team name: NN-MIH).

5. Conclusion

In this study, we proposed DIGRU using patient demographics as the initial value of the hidden state, and tried to predict sepsis from data obtained in ICU. In the proposed model, the utility score was directly optimized by custom loss function and two-step training was performed.

In the future, we would like to try semi-supervised learning using unlabeled external data to improve prediction.

Acknowledgments

We would like to thank Nami Saito and Tatsuki Koga for fruitful discussions.

References

[1] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.

[2] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).

[3] Graves, Alex, Abdel-rahman Mohamed, and Geoffrey Hinton. "Speech recognition with deep recurrent neural networks." 2013 IEEE international conference on acoustics, speech and signal processing. IEEE, 2013.

[4] Lai, Siwei, et al. "Recurrent convolutional neural networks for text classification." Twenty-ninth AAAI conference on artificial intelligence. 2015.

[5] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.

[6] Chung, Junyoung, et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling." arXiv preprint arXiv:1412.3555 (2014).

[7] Reyna, M. A., et al. "M. Brandon Westover MB, Nemati S, Clifford GD, Sharma A. Early prediction of sepsis from clinical data: the PhysioNet/Computing in Cardiology Challenge 2019." Critical Care Medicine (2019).

[8] Ba, Jimmy Lei, Jamie Ryan Kiros, and Geoffrey E. Hinton. "Layer normalization." arXiv preprint arXiv:1607.06450 (2016).

[9] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[10] Loshchilov, Ilya, and Frank Hutter. "Decoupled weight decay regularization." arXiv preprint arXiv:1711.05101(2018).

[11] Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." The journal of machine learning research 15.1 (2014): 1929-1958.

Address for correspondence:

Jun Seita
Nihonbashi 1-chome Mitsui Building, 15th floor 1-4-1 Nihonbashi, Chuo-ku, Tokyo 103-0027, Japan
jun.seita@riken.jp