Hybrid Feature Learning Using Autoencoders for Early Prediction of Sepsis

Jia Yao[1], Ming Lun Ong[2], Kar Kin Mun[3], Shiyu Liu[4], Mehul Motani[5]

[1] Department of Electrical and Computer Engineering
National University of Singapore, Singapore

Abstract

Sepsis is a critical disease that often lead to problems of morbidity and mortality. The risk of sepsis increases when it is not able to be diagnosed at the early stage. In this paper, we propose a hybrid feature learning model, combining both spatial and temporal autoencoders, to learn spatio-temporal representations in an unsupervised manner for the early stage prediction of sepsis. The idea is that the hybrid model should be able to learn good feature representations which leads to better classification results than standard feature learning approaches. We use PhysioNet cardiology challenge dataset to test our hybrid model and compare the results with feature sets learned from baseline feature learning algorithms, across a diverse set of classifiers. The hybrid model consistently outperforms the baseline algorithms for all classifiers, with improvements by 5%.

1. Introduction

Sepsis is a critical disease which may cause patients to suffered from significant problems such as morbidity and mortality. Patients in intensive-care units (ICUs) are more susceptible to develop sepsis, as they have a higher chance to be infected due to organ failures and tissue damages [1]. Diagnosis time plays a crucial part in fighting against the problem of sepsis. The risk of sepsis increases when it is not able to be diagnosed at the early stage. Developing an effective method for early sepsis prediction is therefore critical.

In this paper, we propose a hybrid feature learning model containing spatial and temporal autoencoders to learn deep feature representations from respective domains of high-dimensional and multi-channel time series patient data for early stage prediction of sepsis. The proposed hybrid model is unsupervised in nature, and it is designed to learn signatures in patient's health data from both spatial and temporal domains with different capturing orders. By stacking the spatial and temporal autoencoders in the model, the proposed hybrid model is able to effectively identify the patterns and recognize the spatio-temporal dependencies in the patient data, and thus learn deep feature representations from patient data for better predictive diagnosis of sepsis.

2. Spatio-temperal Feature Learning

Our proposed hybrid spatio-temporal feature learning model consists of a spatial autoencoder (SAE) and a temporal autoencoder (TAE). Both autoencoders can be used to learn certain feature representations from various aspects of multivariate time series data. In this paper, $X \in \mathbb{R}^{\mathbb{S}\times\mathbb{T}}$ is used to represent the raw patient data, where S is the number of channels and T is the number of time stamps. Therefore, we consider X in the form, such as $X = [x_1, x_2, ..., x_\tau]$, where $x_\tau \in \mathbb{R}^{\mathbb{S}}, \forall\tau = 1, ..., \mathrm{T}$. x_τ includes all channel records at time stamp τ.

2.1. Spatial Autoencoder

The SAE is a standard autoencoder constructed with multi-layer neural networks [2], [3], consisting of an encoder and a decoder. The encoder Spatial-E can have multiple hidden layers where each layer extracts a compact representation from the previous hidden layer. Through the encoder Spatial-E, the input data at each time stamp $x_\tau \in \mathbb{R}^{\mathbb{S}}$ is mapped into a compact representation $Y_\tau \in \mathbb{R}^{\mathbb{S}'}$ through a deterministic mapping:

$$Y_\tau = f_{\boldsymbol{\theta}^{se}}(x_\tau) = s(\mathbf{W}\mathbf{x}_\tau + \mathbf{b}), \tag{1}$$

parameterized by $\boldsymbol{\theta}^{\mathbf{se}} = (\mathbf{W}, \mathbf{b})$, where the weight matrix $\mathbf{W} \in \mathbb{R}^{\mathbb{S}'\times\mathbb{S}}$ and the bias vector $\mathbf{b} \in \mathbb{R}^{\mathbb{S}'}$. $s(.)$ denotes an activation function, which performs a non-linear transformation of the given data. Through the spatial encoder, $\boldsymbol{Y} = [Y_1, Y_2, ..., Y_\tau]$, where $Y_\tau \in \mathbb{R}^{\mathbb{S}'}, \forall\tau = 1, ..., \mathrm{T}$, is the spatial representations of the raw input data X. The decoder Spatial-D has the symmetric structure of the encoder Spatial-E, where the compact representations generated by the encoder Spatial-E at each time stamp τ (i.e., $Y_\tau \in \mathbb{R}^{\mathbb{S}'}$) are reconstructed back to the input data $\hat{x}_\tau \in \mathbb{R}^{\mathbb{S}}$ trough a deterministic mapping:

$$\hat{x}_\tau = f_{\boldsymbol{\theta}^{se'}}(Y_\tau) = s(\mathbf{W}'\mathbf{Y}_\tau + \mathbf{b}'), \tag{2}$$

which parameterized by $\boldsymbol{\theta}^{\mathbf{se}'} = (\mathbf{W}', \mathbf{b}')$, where $\mathbf{W}' \in \mathbb{R}^{\mathbb{S}\times\mathbb{S}'}$ and $\mathbf{b}' \in \mathbb{R}^{\mathbb{S}}$.

The training process of SAE works as follow. A multivariate time series patient data X is first sliced into x_τ, (i.e., x_τ is the patient's multi-channel measurements recorded at time stamp τ), where $\tau = 1, ..., T$. Each x_τ is used to train SAE and update its parameters $\boldsymbol{\theta}^{se}$ and $\boldsymbol{\theta}^{se'}$. In our study, P number of patients data are involved in the training process. Each patient has a length of T measurements recorded in total. Therefore, the SAE is trained by $P \times T$ number of data samples iteratively.

In this paper, both Spatial-E and Spatial-D contain only one hidden layer with a total of S' and S neurons, respectively. The rectified linear unit and sigmoid are used as the activation functions for the Spatial-E and Spatial-D, respectively. Both parameters $\boldsymbol{\theta}^{se}$ and $\boldsymbol{\theta}^{se'}$ are updated and trained to minimize the average reconstruction error between x_τ and $\hat{x}_\tau$ [3], [4], which is measured by the mean squared error (MSE) loss function $L(x_\tau, \hat{x}_\tau)$ over the training data. The SAE is optimized using Adam [5].

2.2. Temporal Autoencoder

The TAE in this paper is constructed with long short-term memory (LSTM) cell [6], which is a special type of recurrent neuron networks (RNN). Each LSTM cell contains four gates, namely forget gate (f), input gate (i), update gate (u) and output gate (o). Each gate takes in both the current input (e.g., x_τ) and the output from the hidden state (e.g., $h_{\tau-1}$) of the previous LSTM cell. The mathematical operations within each LSTM cell at time stamp τ are computed based on the following:

$$f_\tau = A_f(W_f[h_{\tau-1}, x_\tau] + b_f) \quad (3)$$
$$i_\tau = A_i(W_i[h_{\tau-1}, x_\tau] + b_i) \quad (4)$$
$$u_\tau = A_u(W_u[h_{\tau-1}, x_\tau] + b_u) \quad (5)$$
$$C_\tau = C_{\tau-1} * f_\tau + i_\tau * u_\tau \quad (6)$$
$$o_\tau = A_o(W_o[h_{\tau-1}, x_\tau] + b_o) \quad (7)$$
$$h_\tau = o_\tau * \tanh(C_\tau). \quad (8)$$

Each gate in an encoder cell LSTM-E of TAE has one hidden layer with a total number of D units, which are used to learn and extract temporal dependencies from the multivariate time series input data X, that has a dimension of $S \times T$ (i.e., S is the number of channel measurements, and T is the length of time stamps). As a result, in (3)-(8), $f_\tau \in \mathbb{R}^{\mathbb{D}}$, $i_\tau \in \mathbb{R}^{\mathbb{D}}$, $u_\tau \in \mathbb{R}^{\mathbb{D}}$ and $o_\tau \in \mathbb{R}^{\mathbb{D}}$ are the output signals from the four gates, and A_f, A_i, A_u and A_o are the activation functions at each gate respectively. Sigmoid function is used for A_f, A_i and A_o, and hyperbolic function is used for A_u. The cell state ($C_\tau \in \mathbb{R}^{\mathbb{D}}$) of the current LSTM cell is updated based on the previous cell state, and controlled by the outputs from the forget gate, input gate and update gate, whereas the hidden state ($h_\tau \in \mathbb{R}^{\mathbb{D}}$) of the current LSTM cell is computed by the signal from the output gate and the activated output of the current cell state. Then, the combined signals, C_τ and h_τ, are the ones which will be passed to the next LSTM cell corresponding to the time stamp $\tau + 1$. $\boldsymbol{\theta}^{te} = \{W_f, b_f, W_i, b_i, W_u, b_u, W_o, b_o\}$ are the parameters for each LSTM-E. In this paper, we assume that the LSTM-E at different time stamps share the same set of parameters.

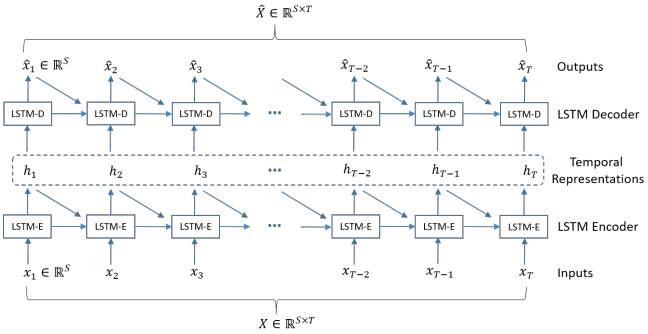

Figure 1. Temporal autoencoder.

Figure 1 illustrates the structure of TAE. We take outputs from the hidden states of every encoder cell LSTM-E in the TAE encoder, and consider them as the temporal feature representations $\boldsymbol{h} \in \mathbb{R}^{\mathbb{D}\times\mathbb{T}}$ of the input data X[7], [8]. In order to train the TAE, a decoder consists of a set of decoder cells LSTM-D, each is with S hidden units in all the gates, are attached after the encoder to map the encoded temporal feature representations $\boldsymbol{h}$ back to the input of TAE, X. Having the same setting as the TAE encoder, all the LSTM-D in the decoder share the same set of parameters, denoted by $\boldsymbol{\theta}^{te'}$.

TAE is trained as follow. The encoder of TAE i.e., LSTM-E, takes in the raw data $X \in \mathbb{R}^{\mathbb{S}\times\mathbb{T}}$ to capture the temporal variations. As a result, the outputs from the hidden states of all the LSTM cells in the encoder of TAE, i.e., $\boldsymbol{h} \in \mathbb{R}^{\mathbb{D}\times\mathbb{T}}$, and $\boldsymbol{h} = [h_1, h_2, ..., h_T]$ is then fed into the decoder of TAE i.e., LSTM-D, to reconstruct the raw data $\hat{X}$ by decoding, where $\hat{X} \in \mathbb{R}^{\mathbb{S}\times\mathbb{T}}$ is the reconstructed signal from the decoder of TAE based on $\boldsymbol{h}$.

The parameters of the TAE, i.e., $\boldsymbol{\theta}^{te}$ and $\boldsymbol{\theta}^{te'}$ are optimized by minimizing the reconstruction error (MSE) via Adam. Similar to the spatial autoencoder, in total we use P number of patients in the training process of TAE. Thus, the parameters of TAE are updated and trained with P data samples iteratively.

2.3. Stacking Spatial and Temporal Autoencoders

As motivated by [9], the two types of autoencoders described above can be stacked together to learn deeper fea-

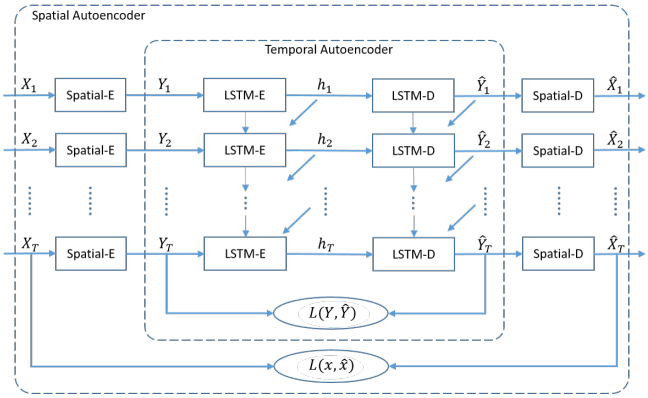

Figure 2. STAE model structure.

ture representations from both spatial and temporal domains. We use STAE to represent the stacked autoencoders which attaches a TAE after a SAE. Figure 2 presents the structure of the STAE model. In STAE model, raw patient data X is passed into a SAE to learn the spatial representation first, denoted as $\boldsymbol{Y}$, and then $\boldsymbol{Y}$ is passed into the next stage, a TAE, to continue learning the temporal representation, denoted as $\boldsymbol{h}$. The autoencoder at each stage of the model is trained and optimized individually. Similar to STAE model, we use TSAE to denote a model that stacks a SAE after a TAE. In TSAE model, the temporal representation $\boldsymbol{h}$ generated by TAE is passed into a SAE to further learn the spatial representation $\boldsymbol{Y}$. To examine the effectiveness of such hybrid stacked autoencoder model in learning the feature representations, single-type autoencoder models (e.g., SAE and TAE) are also constructed as baselines for the comparison purposes.

3. Performance Evaluation

Dataset: Within the PhysioNet Computing in Cardiology Challenge dataset, we note that many features (e.g., SaO2, PaCO2) contain more than 90% missing values and these missing values are imputed with their corresponding latest historical values. In addition, we shift the sepsis label ahead by another 6 hours to perform 12-hour early prediction. Finally, we normalize the dataset from 0 to 1 by features.

Experiment Setup: The dataset is randomly partitioned into 5-folds on the patient-level, where 80% of data is used as training data , and 20% of patients are used for evaluation. Three benchmark classification models: Decision Tree [10], Random Forest [11], Logistic Regression [12] are shortlisted as they are reported to have good performance in the literature [13].

In Table 1, we compare the classification performance using different feature sets learned by various combinations of autoencoders with three shortlisted classifiers respectively. The performances of using the raw data as a feature set with each classifier are also shown for comparison purposes.

Experiment Results: With the random forest classifier, Table 1 shows that the feature set learned by TSAE model achieves the highest classification accuracy among all feature sets learned by different feature learning methods, and it also outperforms the raw data. Using the TSAE feature set, we can classify a patient to have sepsis with 77.2% accuracy, which is about 5.8% and 2.8% higher than using the feature sets extracted from the single-type autoencoder model, i.e., SAE and TAE models for classification respectively. Similar observations can also be seen with decision tree and logistic regression classifiers. Moreover, the AUC-ROC and F1-score achieved by TSAE feature set are generally higher than feature sets learned by other baseline feature learning models and the raw data with all three different classifiers. These observations imply that the TSAE model is more effective in extracting spatio-temporal features from the multivariate time series patient data compared to the other models using single-type autoencoder, as well as directly using the raw patient data as features.

The reason of TSAE model generally outperforms SAE model is that the temporal autoencoder in TSAE can also capture spatial correlations of the raw data. When $x_\tau \in \mathbb{R}^{\mathbb{S}}$ and $h_{\tau-1} \in \mathbb{R}^{\mathbb{D}}$ are passed into the gates in LSTM, the total number of inputs are summarized and re-represented by D hidden units in the network. Therefore, the output from the hidden state of a LSTM cell at time stamp τ, $h_\tau \in \mathbb{R}^{\mathbb{D}}$ learns some spatial dependencies in x_τ. Thus, the structure of TSAE seems similar to stacking one temporal autoencoder and two spatial autoencoders together (i.e., one strong and one weak spatial autoencoders). Such structure has a stronger ability to capture the spatial correlations. Moreover, the temporal autoencoder in TSAE captures the temporal correlations whereas SAE does not have abilities to capture the temporal correlations. Therefore, TSAE has a better classification performance than SAE feature sets in overall.

Similar reasons applied to the comparison between TSAE and TAE. The spatial autoencoder in TSAE captures spatial correlations, and the temporal autoencoder has ability to learn certain spatial correlations at the same time, whereas TAE only has limited capabilities to learn spatial correlations to a certain level. It means that TSAE is more effective in learning useful feature representations from both spatial and temporal domains as compared to the structure constructed only by a single-type autoencoder.

While combining the SAE model and TAE model to construct a hybrid feature learning model (i.e., STAE and TSAE), the order of stacked autoencoders does affect the classification performance of the extracted feature sets. With the PhysioNet Computing in Cardiology Challenge

	Decision Tree			Random Forest			Logistic Regression		
	Acc	F1	AUC-ROC	Acc	F1	AUC-ROC	Acc	F1	AUC-ROC
No Autoencoder	0.659	0.236	0.529	0.741	0.153	0.531	0.645	0.247	0.511
SAE	0.668	0.244	0.515	0.730	0.166	0.522	0.597	0.277	0.541
TAE	**0.679**	0.264	**0.541**	0.751	0.173	0.511	**0.604**	0.285	0.534
STAE	0.668	0.268	0.527	0.743	0.168	0.509	0.602	0.283	0.544
TSAE	0.674	**0.276**	0.525	**0.772**	**0.192**	**0.533**	0.593	**0.313**	**0.566**

Table 1. Performance of Benchmark Models Before and After Various Types of Autoencoder

dataset, feature set extracted by TSAE model is better than the feature set extracted by STAE model in classification accuracies with decision tree and random forest classifiers respectively. Similar performance patterns can also be seen in AUC-ROC and F1-scores with all three classifiers. In this case, we suspect that the order of stacking different types of the autoencoders is data-centric, which is closely dependent on the amount of spatial and temporal information carried in the data. By computing the correlations in the raw data, we obtain the spatial correlations of 0.37 and the temporal correlations of 0.95. We find out that when the data set has relatively higher temporal correlations, it is more effective to firstly extract its temporal dependencies then extract its spatial dependencies, as learning spatial information would potentially break the initial strong temporal dependencies in the data.

4. Conclusion

In this paper, we demonstrate the effectiveness of hybrid spatio-temperoal feature learning model in the early prediction of sepsis. Moreover, we found that the order of stacking the spatial and temporal autoencoders is determined by the spatial and temporal correlations in the data.

Acknowledgments

This work was supported by the Singapore Ministry of Education under grants WBS R-263-000-D35-114 and WBS R-263-000-D64-114.

References

[1] Singer M, Deutschman CS, Seymour CW, Shankar-Hari M, Annane D, Bauer M, Bellomo R, Bernard GR, Chiche JD, Coopersmith CM, et al. The third international consensus definitions for sepsis and septic shock (sepsis-3). Jama 2016;315(8):801–810.

[2] Vincent P, Larochelle H, Bengio Y, Manzagol PA. Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th International Conference on Machine Learning, ICML '08. ISBN 978-1-60558-205-4, 2008; 1096–1103.

[3] Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol PA. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. J Mach Learn Res December 2010;11:3371–3408. ISSN 1532-4435.

[4] Bell AJ, Sejnowski TJ. An information-maximization approach to blind separation and blind deconvolution. Neural Comput November 1995;7(6):1129–1159. ISSN 0899-7667.

[5] Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv14126980 2014;.

[6] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput November 1997;9(8):1735–1780. ISSN 0899-7667.

[7] Sutskever I, Vinyals O, Le QV. Sequence to sequence learning with neural networks. In Advances in Neural Information Processing Systems (NIPS). 2014; 3104–3112.

[8] Cho K, van Merrienboer B, Bahdanau D, Bengio Y. On the properties of neural machine translation: Encoder-decoder approaches. In Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8), 2014. 2014; .

[9] Patraucean V, Handa A, Cipolla R. Spatio-temporal video autoencoder with differentiable memory. CoRR 2015;abs/1511.06309. URL http://arxiv.org/abs/1511.06309.

[10] Quinlan JR. Induction of decision trees. Mach Learn March 1986;1(1):81–106. ISSN 0885-6125. URL http://dx.doi.org/10.1023/A:1022643204877.

[11] Breiman L. Random forests. Machine Learning 2001; 45(1):5–32.

[12] Wright RE. Logistic regressions. Reading and understanding multivariate statistics 2004;217–244.

[13] Shaza.M.Abd.Elrahman, Ajith.Abraham. A review of class imbalance problem. Journal of Network and Innovative Computing 2013;1:332–340.

Address for correspondence:

Ming Lun Ong
4 Engineering Drive 3, E4-06-12,
Communication Lab, Singapore 117583
ongminglun@u.nus.edu