Attention-based LSTM: A Machine Leanring Approach for Automatic Sleep Stages Classification

Junbin Huang[1], Pengcheng Wang[2], Cong Xie[3], Zehan Zhang[4], Donglai Sun[5]

[1,2,3,4,5]Maxtropy Technology, Shanghai 201199, China
{*huangjunbin,wangpengcheng,xiecong,zhangzehan,sundonglai*}@*maxtropy.com*

***Abstract*—The abstract goes here.**

***Index Terms*—IEEE, IEEEtran, journal, LaTeX, paper, template.**

I. Introduction

THIS demo file is intended to serve as a "starter file" for IEEE journal papers produced under LaTeX using IEEEtran.cls version 1.8b and later. I wish you the best of success.

Such a method has been applied by several groups

Most of the proposed sleep stage classification models belong to a family of model selection, with a certain feature set for the PSG data and a general model like Support Vector Machine or Random Forest.

The potential issue with this feature selection approach is that the features selected need to compress all the necessary information of the specific bio-information signals.

A. Subsection Heading Here

Subsection text here.

1) Subsubsection Heading Here: Subsubsection text here.

II. Methods

A. Attention Mechanism

Nueral processing mechanisms involving attention have been deeply investigated in Neuroscience and Computational Neuroscience [1, 2].As the mental activity perfromed by human, attention is definded as an ability of focusing on specific subsets of the received information despite of their position. By applying this mechanism, neural network can learn as what human being can do[3, 4].

For time-sequence data, learning a soft alignment between the input and the output improves performance [4]. Attention-based model are mainly consiss of the encoder-decoder sequence with fixed length of inner representation. The encoder is used to represent important features of the input sequence with fixed outputs, and the decoder is built to generate the output according to the results of the encoder. Both of the encoder and the decoder are comprised of one or more RNN layers [5]. An attention layer between the encoder and the decoder helps the system to mining and select profounder relationship between the representation of encoder and the prediction of decoder. By keeping each output of the encoder, training to focus on subsets of them selectively and then relink them to the output of decoder, the attention layer frees the model from the loss of compressing features into a fixed length vector and the training of the network will concentrate more on the most important parts [4] .

In previously proposed feature-selection-based model, the goal of the sleep stage classifier is to learn and represent the possibility distribution over output prediction y conditioned on the input feature set X, i.e. $P(y|X)$ [5].. By combining with the attention model, the RNN model will be able to learn the distribution of each output y_n in condition with the previous epochs of the sleep stage, i.e. $y_{i<n}$, and the input feature stream, $X = x_1, x_2, X_3, ..., x_m$:

$$P(y|X) = \prod_n P(y_n|X, y_{i<n}) \tag{1}$$

With the attention mechanism, the conditional probability in Eq.(1) can be rewrited as:

$$\begin{aligned} P(y|X) &= \prod_n P(\ y_n|\ \{y_1, ..., y_{n-1}\}, C_n) \\ &= g(y_{n-1}, s_n, C_n), \end{aligned} \tag{2}$$

where g is a nonliear function used to represent the posibility of y_n, and s_n is the hidden state of th multi-layered RNN. The goal of the attention module is then to derive the features of encoder in s_n which need to be attended to for the next output of decoder [5]. The goal of the encoder is to represent the input with each state s_n contains information about the whole input sequence and the previous output of decoder y_{n-1} to produce a fixed-demensional context vector, C_n.

The context vector is an extraction with the most relevant information in the hidden states seqeuence and the previous output which are used to generate the next output label. It can be calculated as an weighted sum:

$$\begin{aligned} C_i &= f(y_{i-1}, s_i) \\ &= \sum_j \alpha_{ij} s_j \\ &= \sum_j e(y_{i-1}, s_j) s_j \end{aligned} , \tag{3}$$

where the weight coefficient e is a non-linear function with respect to the previous output y_{i-1} and the hidden states s_j [4]. It can be seen as an alignment model which scores the relationship between the j_{th} input and the i_{th} output.

Though this mechanism increase the computing burden, the model can be more purposeful and perform better. Furthermore, the attention layer can tell us what the network actually focus on and to what extent it concentrate on specific input-output pairs.

B. Attention-based LSTM Architecture

As Fig. 1 shows, to implement the attention model, we parameterize it as a simple concatenation layer of perceptron to combine the information from hidden state s and the source-side cortex vector C to generate a hidden state as follows [6]:

$$\tilde{h}_n = tanh(\mathbf{W}_c C_n, \mathbf{W}_s s_n), \tag{4}$$

and then feed the output vector $\tilde{h}$ into an single softmax layer. The output of this layer is the weighted sum of the information contained in part of the hidden state stream that the model focus on. The output z represents the relevance for each variable encoded by a layer of Bi-RNN in each time step according to the context C.

Then z is fed through the decoder consist by single layer of LSTM and a multi-layer dense connected perceptron as shown in Fig. 2. The key equation of the proposed network are described below:

$$\begin{aligned} s_t &= f_{Bi-LSTM}(X_{input})) \\ z_t &= f_{Attention}(Sum(\alpha \cdot S_i)) \\ &= f_{Attention}(Sum(softmax(C, s), s)) \\ h_t &= f_{LSTM}(z_t, C_t, y_{i<t}) \\ O &= y_t = softmax(h_t) \end{aligned}$$

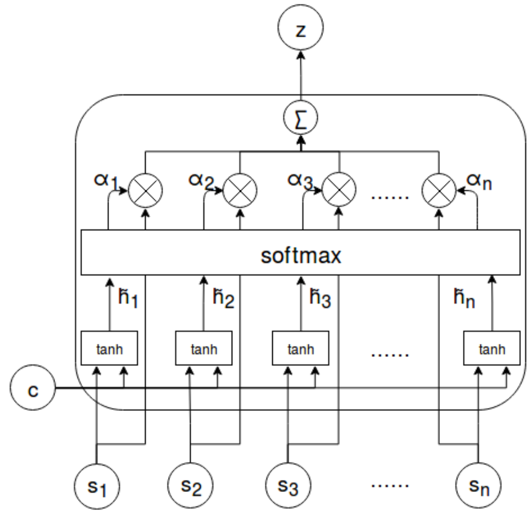
The bias terms are omitted for notational simplicity. The 's' is a sequence of vector with the length as the input X. 'f' denotes some non-linear functions learned by the network, Sum denotes the accumulation, and 'O' denotes the final output of the whole network. Note that for attention mechanism, each time span should have different importance, thus, the proposed non-linear function $f_{Attention}$ is not a static function but should be updated in each time step instead.

III. Experiment

A. Material

In sleep stage classification study, researchers tipycally use polysomnographic (PSG) recorded data as the diagnostic material [5]. The PSG recording experiments are usually conducted in a hospital or sleep center with biological signals such as electroencephalogram (EEG), electrooculogram (EOG), electrocardiogram (ECG) and electromyogram (EMG) of a patient being recorded simulatenously during a whole night experiment [7].

After the data collection procedures, these signals will be split into epochs with 30 seconds. In clinical practice, the classfication for stages of sleep mainly depends on visual observations on epochs according to standards and terminologies established by Rechtschaffen and Kales (R&K) sleep scoring manual [8], or the manual of American Academy of Sleep Medicine (AASM) [9].



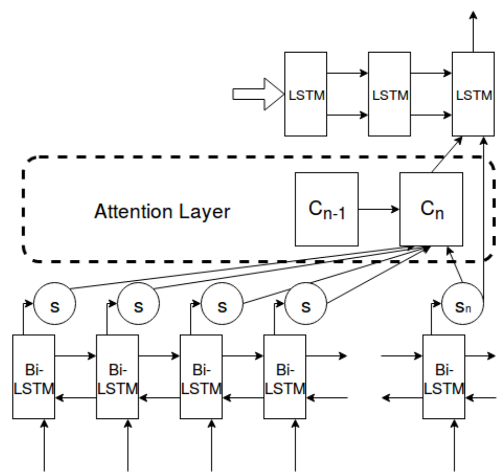


Fig. 1. (Upper) The **attention layer architecture** employed in the network. At each time step n, this layer computes the attentional hidden state on each previous state and then produces and screened weighted summery of the relevance for each input state according to the context vector

Fig. 2. (Lower) The **whole Attention-based network** propsoed in this paper.

.

TABLE I
Description of the train and test set for each fold with respect to each sleep stage after removing noisy epochs (Number of epochs and Ratio[%]

	W	N1	N2	N3	R
Number	5833	4248	8611	3538	3206
Ratio(%)	22.9	16.7	33.9	13.9	12.6

In this investigation, we introduced a dataset contains PSGs of overnight record with 512 Hz of sampling rate recorded from 28 Asian female and male adults. According to recently proposed research [10], there is a trade-off for classification performance among the number of channels, the number of records and spatial extension. The PSG data we investigated includes 6 EEGs, 2 EOG, 4 EMGs 1 ECG and the snore signal. The labels were tagged by several experienced experts according to the AASM guidebook with 5 labels, Awake (W), NREM-1 (N1), NREM-2 (N2), NREM-3 (N3) and REM (R). We firstly separated the records into 30-second long epochs combined with label, excluded the epochs labeled by "?" and then concatenated them into two type of set: one was mixed with respect to epochs, the other was mixed with respect to subjects. Thus we have to kind of dataset, one has 25436 epochs as example shown in Table I; the other has 28 examples with 900 to 1000 epochs. Details about the dataset is attached on the appendix. In this paper, we test The model was tained to minimize the categorical crossentropy with a balanced loss function in order to obtain a relatively impartiality model prediction for all of the sleep stage. What is more, for the purpose of enchancing the discrimination of the normaly under-represented stage, like N1, we increased its weight in the loss function. To test our model, we trained the model with 5-fold cross-validation (CV) with the two datasets respectively for its abilities of learning representations for each sleep stage and the generalization among different people. The model was implemented in *Keras* with a *Tensorflow* backend. As an extension, we also evaluate the classifier's performance with multiple values of class, i.e. $C = 2\ to\ 6$, as our model on both of these two dataset and it all achieve performs of the state-of-the-art methods.

B. Preprocessing

In preprocessing procedure, we filter the EEG signals into five frequency band, alpha, delta, theta, beta and gamma according to previous studies and the AASM manual [7, 9, 11]. Then the signals are combined together again to produce a new single signal with the shape of $(-1, 512 \times 30)$. The '-1' denotes the number of the epochs in each sample.

C. Feature extraction

In this experiment, we trained our models on the features extracted from original PSG data records: time domain and frequency domain features mentioned in previous proposed papers [7, 11, 12].

Specifically, we firstly caculated the power spectral density as the energy of the 5 frequency band for each channel: delta (δ, 1 - 4 Hz), theta (θ, 4 - 8 Hz), alpha (α, 8 - 14 Hz), beta (β 14 - 31 Hz), gamma (γ, 31 - 50 Hz). The ratio (PSD of each band to PSD of the whole) and the relative value (PSD of each band to another band) are also extracted from the PSD result. What is more, the statistic values such as maximum, minimum, mean, standard deviation, skewness and kurtosis are calculated from both the time and frequency domain. Furthermore, the Hjorth features, 95% and 50% of spectral edge frequency and the statistics of them are included as suppplementary features. Finally the feature set contains 770 features with 30 time-step (one second for each without overlap).

Since some of the sleep stages' definition and classification contains the stages of the previous epochs, we included features from 2 epochs before and after the current epoch respectively (30×2 seconds for both side) [10].

D. Training

In the experiments, we used one single machine with Intel E5-2683v2 CPU×2, and 128GB memory, equiped with a Nvidia GeForce GTX 1080 graphics card. We used the recorded data and the devision of dataset as mentioned in $Subsection\ A$.

The model was tained to minimize the categorical crossentropy with a balanced loss function in order to obtain a relatively impartiality model prediction for all of the sleep stage. What is more, for the purpose of enchancing the discrimination of the normaly under-represented stage, like N1, we increased its weight in the loss function. To test our model, we trained the model with 5-fold cross-validation (CV) with the two datasets respectively for its abilities of learning representations for each sleep stage and the generalization among different people. The model was implemented in *Keras* with a *Tensorflow* backend. As an extension, we also evaluate the classifier's performance with multiple values of class, i.e. $C = 2\ to\ 6$, as other previous research [13]. As a comparison, we employed an Gradient Boosting Classifier implemented with *XGBoost* [14] and a 2-layer LSTM network training on the same data.

IV. Conclusion

The conclusion goes here.

Appendix A
Proof of the First Zonklar Equation

Appendix one text goes here.

The authors would like to thank...

References

[1] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[2] R. Desimone and J. Duncan, "Neural mechanisms of selective visual attention," *Annual review of neuroscience*, vol. 18, no. 1, pp. 193–222, 1995.

[3] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, and B. Xu, "Attention-based bidirectional long short-term memory networks for relation classification," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, vol. 2, 2016, pp. 207–212.

[4] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[5] R. Prabhavalkar, T. N. Sainath, B. Li, K. Rao, and N. Jaitly, "An analysis of attention in sequence-to-sequence models,," in *Proc. of Interspeech*, 2017.

[6] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[7] M. Ronzhina, O. Janoušek, J. Kolářová, M. Nováková, P. Honzík, and I. Provazník, "Sleep scoring using artificial neural networks," *Sleep medicine reviews*, vol. 16, no. 3, pp. 251–263, 2012.

[8] A. Rechtschaffen and A. Kales, "A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects," 1968.

[9] R. B. Berry, R. Brooks, C. E. Gamaldo, S. M. Harding, C. Marcus, and B. Vaughn, "The aasm manual for the scoring of sleep and associated events," *Rules, Terminology and Technical Specifications, Darien, Illinois, American Academy of Sleep Medicine*, 2012.

[10] S. Chambon, M. Galtier, P. Arnal, G. Wainrib, and A. Gramfort, "A deep learning architecture for temporal sleep stage classification using multivariate and multimodal time series," *arXiv preprint arXiv:1707.03321*, 2017.

[11] F. Ebrahimi, M. Mikaeili, E. Estrada, and H. Nazeran, "Automatic sleep stage classification based on eeg signals by using neural networks and wavelet packet coefficients," in *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*. IEEE, 2008, pp. 1151–1154.

[12] F. Chapotot and G. Becq, "Automated sleep–wake staging combining robust feature extraction, artificial neural network classification, and flexible decision rules," *International Journal of Adaptive Control and Signal Processing*, vol. 24, no. 5, pp. 409–423, 2010.

[13] T. Nakamura, T. Adjei, Y. Alqurashi, D. Looney, M. J. Morrell, and D. P. Mandic, "Complexity science for sleep stage classification from eeg," in *Neural Networks (IJCNN), 2017 International Joint Conference on*. IEEE, 2017, pp. 4387–4394.

[14] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," pp. 785–794, 2016.