A Study on Automatic Sleep Stage Classification Based on CNN-LSTM

Yang Yang[1,2]
1(School of Information Science and Engineering, Shandong Normal University, Ji'nan 250014, China)
2(Shandong Provincial Key Laboratory for Distributed Computer Software Novel

Xiangwei Zheng[1,2]
1(School of Information Science and Engineering, Shandong Normal University, Ji'nan 250014, China)
2(Shandong Provincial Key Laboratory for Distributed Computer Software Novel

Feng Yuan[3,4]
Technology, Ji'nan 250014, China)
3(Key Laboratory of TCM Data Cloud Service in Universities of Shandong (Shandong Management University)，Jinan250357,China)
4(School of Information Engineering，Shandong Management University，Jinan 250357,China)

ABSTRACT

Automatic Sleep Stage Classification (ASSC) plays an important role in the diagnosis of sleep related diseases. However, due to the complexity of mathematical modelling, ASSC has many difficulties. At the same time, the rapid fluctuations between the adjacent sleep stages make it difficult to extract features, resulting in an inaccurate classification of a period of electroencephalogram (EEG) sleep. In order to solve the above problems, this paper proposes a sleep stage classification method based on convolutional neural network and long-term short-term memory network (CNN-LSTM). The method applies CNN to extract spatial features from the original data and LSTM to extract temporal features and adopt softmax to classify these features. To verify the proposed method, we tested it on a public data set called ISRUC-Sleep and compared it with several state-of-the-art methods. The experimental results show that the proposed method significantly improves the accuracy of sleep staging and achieves better results.

CCS CONCEPTS

• Human-centered computing ~ Collaborative • Social computing

KEYWORDS

classification convolutional neural network, feature extraction, long-term and short-term memory network, sleep staging.

ACM Reference format:

Yang Yang, Xiangwei Zheng and Feng Yuan. A Study on Automatic Sleep Stage Classification Based on CNN-LSTM *In proceeding of ICCSE'18, Singapoer, July, 2018,* 5 pages.

ICCSE'18, July, 2018, Singapoer

DOI: 10.1145/3265689.3265693

1 INTRODUCTION

Sleep is an important process of the body's recovery, integration and consolidation of memory, and it is an integral part of health [1, 2]. Reasonable sleep staging is the basis for studying sleep quality and diagnosing sleep disorders. According to the R&K sleep staging rules proposed by Rechtchaffen and Kales in 1968, the sleep process can be divided into the awakening period (Awake), the non-rapid eye movement sleep (NREM) and the rapid eye movement sleep (REM) [3]. Furthermore, the non-rapid eye movement consists of four stages, namely S1~S4. Due to the very similar characteristics of S3 and S4, the American Academy of Sleep Medicine (AASM) combined them into deep sleep stages.

However, the regular use of R&K standards has revealed many limitations. As a result, a more advanced way to replace R&K became the new golden rule. The scoring criteria for the AASM have been issued, which means the revision of the guidelines and the application of new terms. The AASM scoring criteria were issued, which meant the revision of the guidelines and the application of new terms. The new guidelines changed the sleep stage as follows: N1, N2, and N3 replaced S1-S4 (S3 and S4 merged into N3) and MT was abolished. The definition of various stages of sleep has been slightly expanded based on the R&K standards.

There are two main categories of automatic sleep staging: (1) Shallow learning: In 1992, Roberts [4] et al. applied artificial neural networks to the field of sleep staging for the first time, opening a new chapter for sleep staging. In the 1990s, shallow learning algorithms such as Support Vector Machine, Boosting, and Maximum Entropy Method (as Logistics Regression, LR) were successively proposed. (2) Deep learning: the concept of deep learning originates from artificial neural network, contains multilayer perceptron, has excellent feature learning ability, and can form more abstract and essential highly level features through low layer characteristics.

Initializing layer by layer through unsupervised learning in the training process solves the problem of artificial neural network training, such as the Deep Belief Network (DBN). It is noteworthy that sleep data as a timing signal, sleep staging in order to achieve better results to take into account the timing correlation between the information before and after the signal, and LSTM-RNN can better mine the sleep sequence time information, so this article uses LSTM-RNN.

The remainder is organized as follow. In the second part we introduced the application of CNN and RNN in recent years and in the next part we introduced our model and our training algorithm. In last part we conducted two sets of experiments to verify the validity of our model.

2 RELATED WORK

The use of sleep stages classification has been investigated for several years and has already developed a number of automatic sleep stages methods. The following literatures show several successful cases for sleep stages classification: K-means Clustering, artificial neural network, random forest, support vector machine (SVM), spectral analysis, nonlinear feature analysis and so on.

CNN has attracted much attention for several years. In the past few decades, there has been a lot of research on the use of CNN for classification of sleep stages. For example, some scientists have used CNN to diagnose human-related diseases. Morabito [5] et al. used CNN to deeply classify EEG patterns of Alzheimer's disease (mild cognitive impairment, MCI) and age-matched healthy controls (HC) of Alzheimer's disease (AD). Tsinalis [6] et al. used CNN automated sleep stage scoring based on single-channel EEG learning task-specific filters to classify them without using previous domain knowledge. Although the CNN architecture has performed very well in the classification field, some scientists have optimized the CNN architecture in order to distribute the classification tasks. For better sleep stage classification, Chamnon et al. In 2011, Cecotti [7] et al. The CNN is widely used in the field of object recognition [8], image segmentation [9]. Although there is recently a small group in using CNN for EEG classification [10,11], but there is still sustainable research value.

However, only using convolutional neural networks for feature extraction does not provide a good extraction of time characteristics in EEG. Therefore, the experiment will be conducted using the combination of CNN and LSTM.

3 ASSC BASED ON CNN-LSTM

3.1 Design of CNN-LSTM

EEG signals are signals that have temporal and spatial characteristics. Therefore, a new CNN-LSTM structure has been specifically designed in this paper.

The CNN-LSTM structure is divided into representational learning and residual sequence learning and composed of 9 layers networks. The first layer is the input layer; the second layer to the fourth layer constitute the feature extraction part, which mainly extracts the spatial characteristics; the fifth layer is the fully connected layer, the sixth layer to the ninth layer encode time characteristics, and finally entered softmax [21,22] classifier for classification. The specific network structure and parameters are shown in Figure 1.

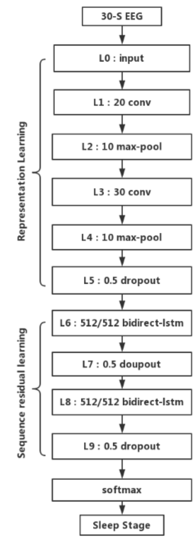


Figure 1: CNN-LSTM architecture

For the representational learning part, the convolutional neural network training process mainly uses the back-propagation algorithm that is input training data to calculate the activation value of each neuron before, and then calculate the error in reverse, and find each weight and bias for the error. Set the gradient and adjust each weight and deviation accordingly. Next we will introduce the propagation process in detail.

The propagation process starts with the input layer. The input layer is designed to convert the data into a standard format$N * T$, where N stands for the number of data channels and T stands for the time sampling points in each channels. Next, the propagation process is resihapng. Following the process, the input matrix is propagated through $L_1$to L_4 by a pair of convolution operation and max-pooling operation. That is to say the input data EEG signals after reshaping firstly convolve with convolutional kernel matrix and then it will be put into a max-pooling layer. Finally, the full-connected layer transfers the matrix from previous layer to output layer.

3.2 Propagation

More specifically, the propagation process mainly includes two processes which are forward-propagation and back-propagation. The forward propagation is from L_1 to L_4 . L_1 and L_3 are convolutional processes which are consisted of different numbers of output maps. The process conducts the "convolutional" operation between the input matrix from the previous layer and the convolution kernel matrix, and travels with the non-linear conversion to get feature maps. So the output of the map m at level l can use the following mathematical equation (1).

$$x_m^l = f\left(\sum_{k=0}^{k=n} \sum_{i=0,j=0}^{i=N,j=SF*TS} x_{SF*TS*i,J+n}^{l-1} w_{m,i,j}^l + b_m^l\right) \quad (1)$$

Where $x^{l-1}_{SF*TS*i}$ is the input of our CNN model, $0 \le i < N, 0 \le j < SF*TS$, N, SF, and TS stand for the number of EEG channels, signal frequency acquisition and the time segment, respectively. In addition, n stands for the neurons in M filters, b^l_m is the set of weights at level l and f is a tanh function.

The layer L_2 and L_4 are the pooling layer which can greatly reduce the amount of network computing while keeping the number of feature maps unchanged. Its mathematical theory is described as formula (2).

$$x^l_m = f(\beta^l_m D_s(x^{l-1}_m) + b^l_m) \tag{2}$$

Where $D_s()$ is pooling functions; β^l_m and b^l_m is the multiplicative bias and additive bias; f is mathematical function we choose. The back-propagation for L_1 and L_3 is done by using a gradient descent by minimizing the least mean square error, which can reduce the network's propagation errors effectively.

We apply the residual learning framework [23] to design our sequence residual learning part. This part consists of two main components: bidirectional-LSTMs and a shortcut connection.

We employ two layers of bidirectional-LSTMs to learn temporal information such as stage transition rules which sleep experts use to determine the next possible sleep stages based on the previous stages. For instance, the AASM manual suggests that if a subject is in sleep stage N2, continue to score epochs with low amplitude and mixed frequency EEG activity as N2 even though K complexes or sleep spindles are not present. In this case, the bidirectional-LSTMs can learn to remember that it has seen the stage N2 and continue to score successive epochs as N2 if they still detect the low amplitude and mixed frequency EEG activity. Bidirectional-LSTMs extends the LSTM by having two LSTMs process forward and backward input sequences independently [24]. In other words, the outputs from forward and backward LSTMs are not connected to each other. The model is therefore able to exploit information both from the past and the future. We also use peephole connections, in our LSTMs which allow their gating mechanism to inspect their current memory cell before the modification.

We use a shortcut connect to reformulate the computation of this part into a residual function. This enables our model to be able to add temporal information it learns from the previous input sequences into the feature extracted from the CNNs. We also use a fully-connected layer in the shortcut connection to transform the features from the CNNs into a vector that can be added to the output from the LSTMs. This layer performs matrix multiplication with its weight parameters, batch normalization, and applying the ReLU activation sequentially. Formally, suppose there are N features from the CNNs $\{a_1 \dots a_n\}$ arranged sequentially and t = 1...N denotes the time index of 30-s EEG epochs, our sequence residual learning is defined as follows (3)(4)(5):

$$h^f_t, c^f_t = LSTM_{\theta_f}(h^f_{t-1}, c^f_{t-1}, a_t) \tag{3}$$

$$h^b_t, c^b_t = LSTM_{\theta_b}(h^b_{t+1}, c^b_{t+1}, a_t) \tag{4}$$

$$o_t = h^f_t || h^b_t + FC_\theta(a_t) \tag{5}$$

where LSTM represents a function that processes sequences of features at using the two-layers LSTM parameterized by θ_f and θ_b for forward and backward directions; h and c are vectors of hidden and cell states of the LSTMs; h^f_0, c^f_0, h^b_{N+1} and c^b_{N+1} of forward and backward LSTMs are set to zero vectors; FC represents a function that transform features at into a vector that can be added (element-wise) with the concatenated output vector $h^f_t||h^b_t$ from the bidirectional-LSTMs. The specifications of the hidden size of forward and backward LSTMs, and the fully-connected layers can be found in Fig. 1. Each Bidirect-Lstms block shows hidden sizes of forward and backward LSTMs. Each fc block shows a hidden size.

3.3 Classification

The full-connected layer, as the name suggests, is full connected with the front layer and the later output layer. Its function formula is described as formula (6).

$$o^l_m = f\left(\sum_{i=1}^{n} x^{l-1}_m w^l_m + b^l_m\right) \tag{6}$$

Where n is the neuron number of the previous layer and w^l_m is the connection strength with the previous layer.

In our CNN-LSTM, the output layer uses softmax classifier with 5 units to classify the sleep stages. The number of units stands for the sleep stage number that can be classified.

3.4 Training

Algorithm 1: Training process

Step 1: The first step in training is to supervise the pre-training of the model's representation learning section with the training set so that the model does not over-adapt to most sleep stages.

1) We have stacked CNN and softmax together as a pre-model.
2) Then, using the gradient-based small batch optimizer Adam, the pre-models are trained in the rate of learning.
3) At the end of the pre-training, the softmax layer is discarded. The class balance training set is obtained by copying a few sleep stages in the original training set so that all sleep stages have the same number of samples.

Step 2: The second step is to perform a supervised fine-tuning on the whole model with a sequential training set. This step is to encode the stage transition rules into the model as well as to perform necessary adjustments on the pre-trained CNNs.

Specifically, the model is trained with the sequence training set using a mini-batch Adam optimizer with two different learning rates, lr1 and lr2. As the CNNs part has already been pre-trained, we, therefore, use a lower learning rate lr1 for the CNNs part and a higher learning rate lr2 for the sequence residual learning part, and a softmax layer. We found that when we used the same learning rate to fine-tune the whole network, the pre-trained CNN parameters were excessively adjusted to the sequential data, which were not class-balanced. As a consequence, the model started to over fit to the majority of the sleep stages toward the end of the fine-tuning. Therefore, two different learning rates are used during fine-tuning. Also, we use a heuristic gradient clipping technique to prevent the exploding gradients, which is a well-known problem when training RNNs such as LSTMs [25]. This technique rescales the gradients to smaller values using their global norm whenever they exceed a pre-defined threshold. The sequential training set is

obtained by arranging the original training set sequentially according to time across all subjects.

4 EXPERIMENTS AND RESULTS

4.1 Data and Preprocessing

To show the robustness of our proposed CNN-LSTM architecture, we apply our proposed CNN architecture to a public sleep datasets called ISRUC-Sleep. The datasets consist of the full overnight recordings of 116 subjects with three health status (healthy, sick, under treatment). Each recording contains four EEG channels (i.e., C3_A2, C4_A1, F3_A2 and F4-A1), four EOG channels (i.e., O1-A2, LOC_A2, ROC-A1and O2-A1) and three EMG channels (i.e., X1, X2, and X3) as well as an annotation file with detailed events. The recording rate is 200 Hz. Each 30s epoch is divided into the one of the five sleep stages mentioned above. All night recordings are labeled by two experts according to AASM rules. More detailed can be found in paper [4].

As we all know, the EEG recordings are often interfered by a variety of factors, such as baseline drift, ocular motions, and white noise. Those artifacts do really have a bad influence on the final sleep stage classification, so we need to do some preprocessing work. A 10th order II_R Butterworth band filter is applied to the EEG signals in order to remove the noise and artifacts from EEG signals. In addition, we also use a 12th orders stop-band Butterworth Notch filter to decrease the interfaces. We both plot a sleep data of 5 seconds with 11channels. It can be clearly see that our preprocessing work really remove the artifacts while keeping the basic characteristics of the signal unchanged at the same time, which will have a good effect on the sleep stages classification task.

4.2 Experiments

We conducted two sets of experiments. One group used the convolutional neural network mentioned in the paper [26] to classify the datasets and the other used our CNN-LSTM to classify the datasets.

In experiments, we obtain results using 10-fold cross-validation. Specifically, in each fold we use the recordings of 10 subjects for testing and the recordings of the remaining 106 subjects for training and validation. For each fold we use the recordings from 10 randomly selected subjects as validation and the recordings from the remaining 86 subjects for training. Each experiment is carried out 5 times shuffling the records for training, validation and testing to reduce variance in matric evaluation.

4.3 Evaluation metrics

We use Accuracy (ACC) and Confusion Matrix to evaluate our criteria. The ACC is calculated as follows:

$$\mathrm{ACC} = \frac{\sum_{c-1}^{C} TP_c}{N} \tag{7}$$

Where TP_c is the true value of the C class, C is the number of sleep stages, and C is the total number of test periods. Each column of the confusion matrix represents a prediction category, and the total number of each column represents the number of data predicted to be that category; each row represents the true attribution category of the data, and the total number of data of each row represents the number of data instances of that category.

4.4 Discussion

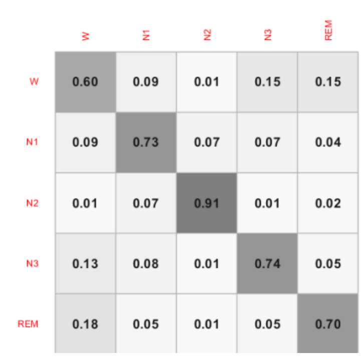
As shown in Table 1, we compare the results with the convolutional neural network in [26]. Compared to a convolutional neural network alone, our average accuracy can be improved by 1.1%.Specifically, we have a significant improvement in the classification accuracy of the W、N2、N3, but there is a slight decrease in the accuracy of the N1 and REM. Raising the classification accuracy of these two periods will be the goal of our next work.

Table 1: The AASM staging criteria

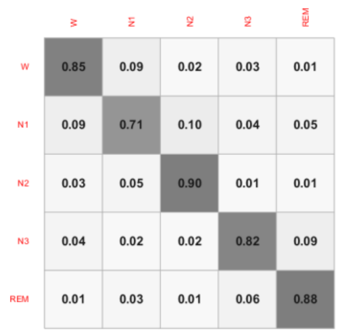
Algorithm model	Mean ACC	Subclass accuracy				
		W	N1	N2	N3	REM
CNN	83	82.34	66.81	85.33	79.72	88.42
CNN-LSTM	84.1	84.61	56.37	90.46	84.34	86.10

As shown in Figure 2, From the results, our network has a significant improvement in the accuracy of the W stage and N2、N3 stage. Compared to CNN, our CNN-LSTM has a better diagonal coefficient.

Specifically, the classification accuracy of W stage was 85%, that of N1 stage was 71%, that of N2 stage was 90%, that of N3 stage was 82%, and that of REM stage was 88%. Compared with CNN, our classification accuracy of W stage increased by 25%, that of N3 stage increased by 8%, and that of REM stage increased by 18%. The classification accuracy of N1 and N2 was slightly lower than that of CNN. This will also be the direction of our work in the future. We hope to improve the accuracy of classification of stage N1 and stage N2, especially stage N1.



(a) CNN's Confusion Matrix



(b) CNN's Confusion Matrix

Figure 2: CNN and CNN-LSTM Confusion Matrix Comparison

5 CONCLUSIONS AND FUTURE WORK

In this paper we propose a new CNN-LSTM structure for multi-channel EEG sleep staging. The entire structure is divided into two parts: representational learning and residual sequence learning, consisting of a total of 9 layers of network. The first layer is the input layer; the second layer to the fourth layer constitute the feature extraction part, which mainly extracts the spatial features; the fifth layer is the fully connected layer, the sixth layer to the ninth layer encode time characteristics, and finally we enter the data into softmax for classification

In order to evaluate the performance of our model, we conducted experiments and finally achieved an average accuracy of 84.1%. This is a 2% improvement over the CNN model in the literature. This proves the superiority of our model's performance.

In the literature, we have found that sample entropy has a certain degree of differentiation in sleep staging [27,28]. In the following work, we plan to input sample entropy as a feature item into our designed model, expecting to improve the accuracy of our model classification.

ACKNOWLEDGMENTS

We are grateful for the support of the National Natural Science Foundation of China (61373149), the National Social Science Foundation of China (16BGL181) and the Natural Science Foundation of Shandong Province (ZR201702130105).

REFERENCES

[1] Fava C, Montagnana M, Favaloro EJ, Guidi GC, Lippi G. Obstructive sleep apnea syndrome and cardiovascular diseases. Semin Thromb Hemost 2011; 37:280-297.

[2] Chambon S, Galtier M N, Arnal P J, et al. A deep learning architecture for temporal sleep stage classification using multivariate and multimodal time series[J]. IEEE Transactions on Neural Systems & Rehabilitation Engineering, 2018, PP (99):1-1.

[3] Ronzhina M, Janoušek O, Kolářová J, et al. Sleep scoring using artificial neural networks [J]. Sleep medicine reviews, 2012, 16(3): 251-263.

[4] Roberts S, Tarassenko L. New method of automated sleep quantification. [J]. Medical & Biological Engineering & Computing, 1992, 30(5):509-517.

[5] Lajnef T, Chaibi S, and Ruby P, et al. Learning machines and sleeping brains: automatic sleep stage classification using decision-tree multi-class support vector machines [J]. Journal of neuroscience methods, 2015, 250: 94-105.

[6] Penzel T, Kantelhardt J W, Becker H F, et al. Detrended fluctuation analysis and spectral analysis of heart rate variability for sleep stage and sleep apnea identification[C]//Computers in Cardiology, 2003. IEEE, 2003: 307-310.

[7] Weiss B, Clemens Z, Bódizs R, et al. Comparison of fractal and power spectral EEG features: Effects of topography and sleep stages [J]. Brain Research Bulletin, 2011, 84(6): 359-375.

[8] Tong N, Lu H, Ruan X, et al. Salient object detection via bootstrap learning[C]//Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on. IEEE, 2015: 1884-1892.

[9] Liping Gao, Bin Shao, Lin Zhu, Tun Lu and Ning Gu. Maintaining Time and Space Consistencies in Hybrid Engineering Environments: Framework and Algorithms. Computers in Industry. Volume. 59 (2008), Issue. 9. pp: 894–904.

[10] Stober S, Cameron D J, Grahn J A. Using Convolutional Neural Networks to Recognize Rhythm Stimuli from Electroencephalography Recordings[C]//Advances in neural information processing systems. 2014: 1449-1457.

[11] Cintas C, Quinto-Sánchez M, Acuña V, et al. Automatic ear detection and feature extraction using Geometric Morphometrics and convolutional neural networks [J]. IET Biometrics, 2016, 6(3): 211-223.

[12] Wulsin D, Blanco J, Mani R, et al. Semi-Supervised Anomaly Detection for EEG Waveforms Using Deep Belief Nets[C]// Ninth International Conference on Machine Learning and Applications. IEEE, 2011:436-441.

[13] Schuster.M and Paliwal.K, 0"Bidirectional recurrent neural networks,"IEEE Trans. Signal Process., vol. 45, no. 11, pp. 2673–2681, 1997.

[14] Pascanu.O, Mikolov.T, and Bengio.Y, "On the difficulty of training Recurrent Neural Networks," arXiv preprint arXiv:1211.5063, 2012.

[15] Tsinalis O, Matthews P M, Guo Y, et al. Automatic Sleep Stage Scoring with Single-Channel EEG Using Convolutional Neural Networks[J]. 2016.