Sleep Stage Classification using Unsupervised Feature Learning

Shashank Rao

Dozee

1 Introduction

Sleep constitutes approximately one third of our lives and has a big impact on health and well-being. Sleep monitoring can aid in the diagnosis of a variety of sleep and psychiatric disorders. Sleep apnea, insomnia, narcolepsy, are all diseases that affect the quality of life of a large amount of population. Our aim is to find a method which is able to classify wake, rapid-eye movement (REM) sleep and non-REM (NREM) light and deep sleep, by analyzing sleep patterns from ECG and respiratory signals on a 30 s epoch basis.

The traditional sleep monitoring, known as polysomnograph (PSG) is done with a wide array of sensors that have to be applied to the body of the patient and that can disturb the sleep of the patient. The increasing interest for automatic sleep stage classification with cardiorespiratory signals is due to the fact that, in contrast to the PSG studies, these signals can be measured using unobtrusive instruments. We are going to use a set of data recorded by a high resolution pressure-sensitive textile sheet, called *Dozee*[1] that uses the Ballistocardiogram (BCG)[2] technique to record signals. This system compared with other techniques has many advantages: is unobtrusive, fits into users familiar sleep environment, and is comfortable.

ECG and respiratory signals consist of multivariate time series characterized by a variety of missing values and heterogeneity between the population. The approach we would like to use is a Deep Belief Network (DBN) to address the data completeness problems, and Long Short-Term Memory (LSTM) network for detection of temporally sequential patterns. Then we will use a softmax function for the classification. This approach has already been used for studying EEG signals[3], but we believe it can provide valuable insights also in the study of BCG.

1.1 Dataset

A total of 8 subjects' data was used for training the networks. It is to be noted that this amount of training data is comparatively small for deep nets. However, keeping in mind the infrastructural limitations, it can be said that this a pretty good training size for prototyping.

The following 8 subjects were used for training:

[1] http://www.dozee.io
[2] http://www.cs.tut.fi/sgn/SSSAG/BCG.htm

- 02122016_Subject G
- 04012017_Subject N
- 04122016_Subject U
- 09012017_Subject N
- 10012017_Subject D
- 12012017_Subject D
- 17022017_Subject M
- 20022017_Subject M

Due to space and computation constraints, the architectures have been tested against one subject : *21022017_Subject7_F*. The performance evaluation for the predictions are explained in Sec 2.

1.2 Advantages of DL for Sleep stage prediction.

There are certain advantages that come with using deep learning modules.

- It is easier and intuitive to introduce non-linearity into the training data using convolution layers.
- Temporal features are captured well due to the sequence training and End-to-End training of LSTM and CNN respectively.
- Training can be stopped mid-way and the state of the network could be used for resuming the training from the same point in future.
- A fully trained model could generate a complete model. This model could be saved for generating predictions or modifying the architecture in case of minor changes in data.

1.3 Limitations

The limitations for the current implementations are:

- Limited usage constraint on GPU instance. The computation time and storage space is highly constrained which leads to frequent deletion of old, processed files.
- Ambiguity in the dataset format. Some parameters like sequence length and no. of samples to consider need to be explored and studied again.
- Inability to store intermediate results. Intermediate results like features learned by CNN, kernel sizes and Gradient Descent scores for each epoch are important to be logged to monitor the performance of the model. However, due to space constraints, none of them were generated.
- The DBN implementation is too simple. It gets the work done but needs more programming to suit our needs.
- Limited cross-validation runs. Due to computation constraints, only the initial parameter values were used and run on single cross-validation run of 5 folds. More CV runs are required to test various combinations and values of parameters.
- As the features learned by the models cannot be visualized, there is no way to reason out why a model has generated a particular prediction. This is where Supervised or combination of Supervised ML could be made use of to better understand the predictions.

1.4 Future Work

I aim to work on the following tasks in the near future to build a better model.

- Rewrite CNN code by following the recently published Stanford's study on ECG-arrhythmia. It contains more specific information on how to use 1-D CNN network for ECG data.
- Save intermediate states and results of CNN so as to prevent loss of data during server outage.
- Write a TensorFlow code for DBN from scratch so that it suits our needs. Features like visualizations, intermediate results and epoch statuses need to be incorporated.
- Perform more cross-validation runs and test the models against various parameter values. CNN's kernel size, stride length, sequence length and number of computation layers need to be set programatically.
- Use *Bayesian Optimization* technique to find the optimal values of hyperparameters.

2 Testing and Evaluation

To evaluate the performance of our architectures, data of 1 subject were used for testing, whereas the remaining 8 subjects were allocated for training. Both datasets were preprocessed with the same preprocessing steps and features were extracted from the training set. Testing was conducted independently for the selected subject using the three architectures. Therefore, we have 3 experimental setups in total to later on help us to take comparative evaluation among the proposed architectures.

The performance of the models was measured by using statistical metrics derived from the confusion matrix. The metrics we computed were classification accuracy, precision, recall, and F-1 score. Ground truth values are required to validate the results. Thus, all of these metrics are computed by comparing the predicted sleep stage with the ground truth sleep labels obtained from NIMHANS.

Once testing was done, the prediction results for both subjects were obtained and subsequently compared to the ground truth values. The comparison of the results are presented in a form of confusion matrix in Tables 1, 2, and 3 for DBN, RNN, and CNN respectively. In these matrices, we could identify the number of samples that were correctly classified or misclassified. However, to make the performance comparison among the architectures in a more clear way, we derive the performance metrics from these confusion matrices, and presented in Table 4.

Table 1. Confusion matrix for DBN

Actual	Predicted			
	1-DEEP	2-LIGHT	3-REM	4-WAKE
1-DEEP	7	0	0	0
2-LIGHT	0	14	2	3
3-REM	0	0	6	1
4-WAKE	1	0	6	10

Table 2. Confusion matrix for RNN

Actual	Predicted			
	1-DEEP	2-LIGHT	3-REM	4-WAKE
1-DEEP	0	4	2	1
2-LIGHT	0	11	2	6
3-REM	0	1	6	0
4-WAKE	0	3	1	13

Table 3. Confusion matrix for CNN

Actual	Predicted			
	1-DEEP	2-LIGHT	3-REM	4-WAKE
1-DEEP	4	1	0	2
2-LIGHT	1	13	1	4
3-REM	0	0	6	1
4-WAKE	1	3	1	12

As presented in Table 4, the full evaluation results on the aforementioned tests clearly show that DBN architecture outperforms both RNN and CNN in terms of accuracy, precision, recall, and F1-score. While CNN has slightly worse performance than the DBN across all the performance metrics. From the scoring performance metrics results, we also observed that RNN came with the worst performance to perform the sleep stage classification.

Table 4. Performance evaluation among DBN, RNN, and CNN across the four scoring performance metrics (accuracy, precision, recall, and F-measure)

Architecture	Accuracy	Precision	Recall	F-Measure
DBN	0.74	0.75	0.61	0.67
RNN	0.60	0.44	0.41	0.42
CNN	0.70	0.70	0.54	0.61

We also assessed the performance of our proposed networks in the sleep stage level. As we show in Table 5, in general from the True Positive Rate, the most correctly classified sleep stage was 4-WAKE with around 83%, 60%, and 80% of the stage correctly classified for DBN, RNN, and CNN respectively. Our proposed architectures were also good enough at classifying 2-LIGHT with a slight worse rate compared to 4-WAKE. On the other hand, the networks performed worst in 1-DEEP and 3-REM. This is likely the case because both stages are in the phase of transition between 1-DEEP to 2-LIGHT or 3-REM to 4-WAKE. Thereby, the networks found 1-DEEP and 2-LIGHT are quite similar as well as 3-REM and 4-WAKE.

Table 5. True Positive Rate and False Positive Rate comparison among DBN, RNN, and CNN in each of sleep stage

Architecture	True Positive Rate					False Positive Rate				
	DEEP	LIGHT	REM	WAKE	AVG	DEEP	LIGHT	REM	WAKE	AVG
DBN	0.368	0.636	0.600	0.833	0.610	0.032	0.000	0.205	0.129	0.092
RNN	0.000	0.733	0.300	0.591	0.406	0.000	0.296	0.172	0.292	0.190
CNN	0.286	0.722	0.333	0.800	0.535	0.061	0.154	0.065	0.233	0.128

References

1. Stephen J. Redmond and Conor Heneghan, ”Cardiorespiratory-based sleep staging in subjects with obstructive sleep apnea”, 2006 IEEE Trans. Biomed. Eng. 53 48596
2. Walter Karlen, Claudio Mattiussi, and Dario Floreano. ”Sleep and Wake Classification With ECG and Respiratory Effort Signals”, IEEE Transactions on biomedical circuits and systems, Vol.3, No.2, April 2009.
3. Hao Dong, Akara Supratak, Wei Pan, Chao Wu, Paul M. Matthews and Yike Guo. ”Mixed Neural Network Approach for Temporal Sleep Stage Classification”, MANUSCRIPT Friday 21st October, 2016.
4. Giri, Endang Purnama, Mohamad Ivan Fanany, and Aniati Murni Arymurthy. ”Combining Generative and Discriminative Neural Networks for Sleep Stages Classification.” arXiv preprint arXiv:1610.01741 (2016).
5. Lngkvist, Martin, Lars Karlsson, and Amy Loutfi. ”Sleep stage classification using unsupervised feature learning.” Advances in Artificial Neural Systems 2012 (2012): 5.
6. Samy, L., Huang, M.C., Liu, J.J., Xu, W. and Sarrafzadeh, M., 2014. Unobtrusive sleep stage identification using a pressure-sensitive bed sheet. IEEE Sensors Journal, 14(7), pp.2092-2101.
7. J. Zhang, Y. Wu, J. Bai, and F. Chen. Automatic sleep stage classification based on sparse deep belief net and combination of multiple classifiers. Sage Journals, 14:1-9, 2015
8. Caffeine, Food, Alcohol, Smoking and Sleep. (2017). Sleep Health Foundation. Retrieved 16 April 2017, from http://www.sleephealthfoundation.org.au/fact-sheets-a-z/262-caffeine-food-alcohol-smoking-and-sleep.html

9. Serkan Kiranyaz, Turker Ince, and Moncef Gabbouj. Real-Time Patient-Specific ECG Classification by 1-D Convolutional Neural Networks. IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, VOL. 63, NO. 3, MARCH 2016.
10. Grasp-and-Lift EEG Detection Winners' Interview: 3rd place, Team HEDJ. (2017). No Free Hunch. Retrieved 2 June 2017, from http://blog.kaggle.com/2015/10/05/grasp-and-lift-eeg-detection-winners-interview-3rd-place-team-hedj/
11. Understanding LSTM Networks – colah's blog. (2017). Colah.github.io. Retrieved 2 June 2017, from http://colah.github.io/posts/2015-08-Understanding-LSTMs/