Using Auxiliary Loss to Improve Sleep Arousal Detection With Neural Network

Bálint Varga, Márton Görög, Péter Hajas

Abstract

Our pipeline consists of a hand-crafted preprocessor and a neural network classifier. We applied transformations on the physiologic signals to gain features in both time- and frequency domains. The proposed algorithm was trained on 994 annotated records of polysomnographic signals. Most of the features were generated from the EEG signal such as power spectral density, and entropy. We extracted features from the EOG, EMG, airflow, and ECG signals too. All the features were normalized.

These 68 features were resampled in 21 non-continuous moments around the current timestamp, and fed into a 3-layer neural network in order to assign a probability of arousal at each second. Arousal samples were enriched during training to battle data imbalance. Additional (auxiliary) losses can guide the network to learn high-level concepts, even though they will not be evaluated. We used sleep stages as additional training targets, which were easier to learn than arousals despite being multi-class. This approach slightly increased arousal AUPRC.

Our submitted results for the entire test set were evaluated: AUPRC=0.42.

Our 10-fold cross validation results for the AUPRC are the following: [0.47110, 0.41672, 0.44305, 0.42842, 0.44644, 0.47969, 0.45082, 0.49320, 0.45913, 0.41278] averaging 0.450.

1. Introduction

As a contribution to the Physionet/Computing in Cardiology Challenge 2018 [1] this paper focuses on the design and implementation of an algorithm which is capable of detecting sleep arousals from polysomnographic signals.

We chose to solve the task with a neural network which can classify each time-point in the input data as either an arousal or a non-arousal region. Instead of feeding the raw 200 Hz polysomnographic data directly to the network our pipeline first preprocesses these signals using domain knowledge henceforth referred to as features. Since the length of the annotated arousal regions are in the magnitude of seconds we chose a resolution of 1 Hz for these intermediate features. We used supervised learning on these feature signals to train a relatively small neural network.

2. Feature extraction

The EEG (electroencephalogram) signals were first analysed in the frequency domain since each sleep stage is characterized by a specific frequency band [2]. Five features were extracted from each EEG channel in the traditional frequency bands [2] with the help of the Welch method [3] in 9 second length windows. Initially we used 3 second length windows but the results showed that the classifier is more stable with a longer window selection.

Approximate Entropy (ApEn) was applied to the EEG signals based on the work in [4] with the suggested parameters. It was developed as a measure of system complexity [5]. A high value of ApEn indicates random and unpredictable variation, whereas a low value of ApEn indicates regularity and predictability in a time series [5].

We used the SaO2 level mostly untouched: we rescaled it between 60% and 100% in order to ensure that it reflects to the physiologically relevant range.

We extracted several features from the EOG (electrooculogram) signal based on the results in [6]. In this publication time- and frequency domain features were applied to the 5 second length windows of the digitized EOG data. Their goal was to detect REM sleep stage with appropriately designed features fed to a neural network. In order to obtain higher classification accuracy they applied Sequential Backward Selection on the features. Based on their results we chose six features, namely the form factor, standard deviation, skewness, kurtosis, and the relative energies in two regions: 0 Hz – 2 Hz and 2 Hz – 4 Hz. These statistics were also applied to the abdomen and chin EMG (electromyogram) signals.

The respiratory signal gives valuable information to arousal detection. In [7] an envelope related signal was evaluated: the Respiratory Disturbance Variable (RDV). As suggested, we processed the previously band-pass filtered signals applying 30 second length windows in 10 second steps. The envelope was obtained by Hilbert Transform. The RDV of the given window was calculated as the ratio of the standard deviation and the mean of the

ISSN: 2325-887X DOI: 10.22489/CinC.2018.247

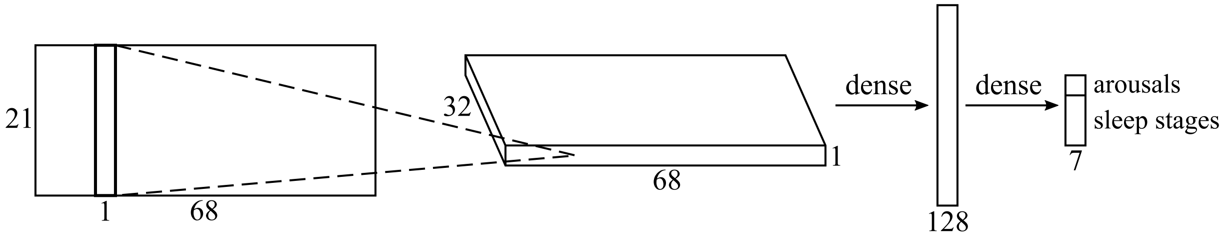

Figure 1. Network architecture.

envelope with the application of a correction factor [7].

We applied Homomorphic Envelope as in [8] for some of the signals such as the EEG channels, abdomen and chin EMGs.

The ECG (electrocardiogram) signal was transformed into a heart rate feature following the basic principles in [9]. The QRS peak detector was modified to filter peak candidates even further by removing peaks whose amplitude did not reach a ratio of a max-pooled value. (A max-pool window of 5 seconds, and a ratio of 0.4 was used).

2.1. Time-slot extraction

Instead of taking continuous chunks of the features, and feeding it to the network, a time-slot extraction is performed first. This was driven by the intuition that we needed a large receptive field. We also wanted to keep the computational and memory costs low, so we avoided dilated convolutions. For this purpose we resampled our 1 Hz feature signals around the examined timestamp in 21 time-slots and continuously concatenated the samples. The sampling is denser at the center and gets gradually sparser at the edges. The whole range of the time-slots provides input from a 2-minute window of the features.

3. Neural network training

3.1. Network structure

The neural network depicted in *Figure 1.* takes a 21 x 68 x 1 (HWC) sized tensor as input, consisting of the timeslot extracted features where H = number of time-slots = 21 and W = number of features = 68. Then a 2D convolution is run with a kernel size of 21 x 1, a filter size of 32, and stride of 21 x 1, resulting in a 1 x 68 x 32 shaped tensor. This tensor is further processed by a fully connected layer with 128 outputs, and finally another fully connected layer with 7 outputs. These 7 outputs make up 2 logits of arousal classification, and 5 logits of sleep stage classification.

We used cross-entropy loss on the classification outputs (one for arousals, and another for the sleep stages), and weighted the losses to balance between the arousal and sleep stage error.

3.2. Auxiliary loss

Unlike some other machine learning methods, neural networks are flexible enough to be able to learn multiple tasks on the same input data at the same time. A major part of the network is shared, only the last layers are separated into independent heads trained with separate objectives. A detailed introduction can be found in [10].

In case one needs to predict multiple targets (like age and sex of a person), and both outputs are necessary, this method is called multi task learning. One can save resources by uniting the two classifiers.

However, if we actually care about only one class of the results (only the sex), but we have additional information which is not provided as an input to the network (age), we may also require the model to output this information, as we actually provide further guidance through the additional loss to the learner (auxiliary loss). Although we will not use this output at inference time, the richer training signal can help build a better representation, and therefore also improve accuracy on the main task.

As the annotations contained sleep-stages, we used this classification as an auxiliary loss. Using it as the network input was not an option, as sleep-stages were not given to the test set.

3.3. Data filtering, data imbalance

The extracted 1 Hz features were slightly preprocessed before being used as training data.

All features were normalized with their mean absolute value to balance patient-differences. We wanted to detect changes during a patient's sleep, and ignore differences between patients. The resulting data was clipped to [-100, 100] to cut outliers which could damage the weights of the neural network. Both normalizing and clipping resulted in better accuracy.

Moments with invalid arousal annotations were skipped, as no evaluation could be made with those. However, moments with invalid sleep stage-annotation were kept.

The data naturally contained many more samples without arousal than with it, which needs to be handled to avoid losing precision. We implemented a data selector, which ensured that at least 25% of the data fed to the network is arousal data, by dropping non-arousal moments when constructing batches.

4. Discussion

4.1. Batch size

Table 1. Batch size effect on AUPRC result.

Batch size	4	10	20	40
AUPRC	0.42549	0.44509	0.45586	0.45969

These tests were run for 50k training iterations, measured on a fixed set of training sequences (validation set) previously unseen by the network. As seen by the table, a batch size of 40 yielded the best results. It is possible that larger batch sizes may provide even further performance benefits, but they are much slower to train, so we kept our batch size at 20 to keep training times reasonable.

4.2. Auxiliary loss weight

Experiments were made to find the best weight of the auxiliary loss. The results are noisy on shorter trainings, but we can conclude that AUPRC deteriorates above 0.5 weight, but may be helpful below (*Figure 2.*).

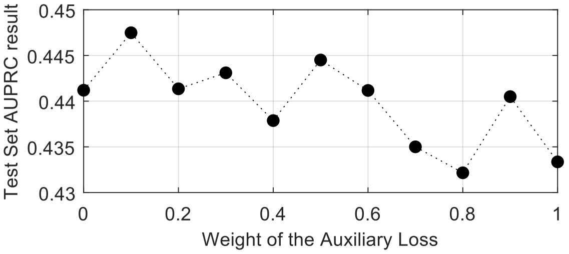

Figure 2. The Effect of the Weight of the Auxiliary Loss.

Additionally, an experiment was made to drop the auxiliary loss during the last 10% of training, when inner representation is constructed, but final weights could support the major target. We measured worse prediction accuracy; therefore, this idea was not used.

4.3. Representation within the neural network

Visualizing weights of the neural network in *Figure 3.* reveals the learned filters. Each column shows weights of a time-slice, while each row is a separate filter. We can find filters which are sensitive to decreasing values in time (first 2 rows). Other mark moments where past and future have higher values and we are at local minimum (8th), one sees a (low) peak in the close future (9th) or simply calculates the average (=sensitive to data with low variance) (18th).

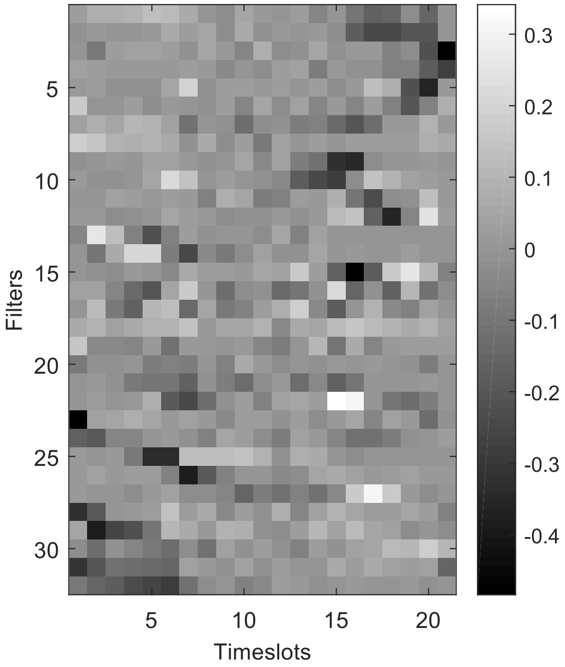

Figure 3. Input filter representation.

This diversity shows rich building blocks for higher layers. Using a higher number of filters did not help training. Higher level weights are not shown as in a fully-connected network they are not easy to interpret.

4.4. Dropout

We have experimented with dropout but found that it did not increase performance. Perhaps when training for more iterations or with a different architecture results would be different.

4.5. Postprocess

Several methods were tried which modified the output arousal probabilities, mostly building on the observation that arousals are annotated in 10 – 30 seconds long blocks. While it seems logical that a lot of arousal-classified samples (in our case the unit was 1 second) reinforce the adjacent lower probability moments, these experiments did not give reliably better results; therefore, they were not used in the final version.

5. Conclusion

We have presented an arousal detector algorithm for polysomnographic data with relatively low resource needs. As the first step, we generate hand-crafted intermediate features using background knowledge. Using this transformed data, a neural network classifies arousal / non-arousal moments.

The evaluation allowed us to classify the data offline — that is after the recording has finished —, however the classifier could easily be modified to work near real-time if a use-case requires it: only the normalization and time-slots would need slight modifications.

References

[1] Ghassemi MM, Moody BE, wei H Lehman L, Song C, Li Q, Sun H, Mark RG, Westover MB, Clifford GD. You Snooze, You Win: the PhysioNet/Computing in Cardiology Challenge 2018. Computing in Cardiology 2018;45:pp 1–4. Maastricht, Netherlands.

[2] Zoubek L, Charbonnier S, Lesecq S, Buguet A, Chapotot F. Feature selection for sleep/wake stages classification using data driven methods. Biomedical Signal Processing and Control 2007;2(3):171–179.

[3] Welch P. The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. IEEE Transactions on Audio and Electroacoustics June 1967;15(2):70–73. ISSN 0018-9278.

[4] Burioka N, Miyata M, Cornélissen G, Halberg F, Takeshima T, Kaplan DT, Suyama H, Endo M, Maegaki Y, Nomura T, Tomita Y, Nakashima K, Shimizu E. Approximate entropy in the electroencephalogram during wake and sleep. Clinical EEG and Neuroscience 2005;36(1):21–24.

[5] Pincus SM. Approximate entropy as a measure of system complexity. Proceedings of the National Academy of Sciences 1991;88(6):2297–2301. ISSN 0027-8424.

[6] Coskun A, Ozsen S, Yucelbas S, Yucelbas C, Tezel G, Kuccukturk S, Yosunkaya S. Detection of rem in sleep eog signals. Indian Journal of Science and Technology 2016; 9(25). ISSN 0974 -5645.

[7] A Díaz J, Arancibia J, Bassi A, Vivaldi E. Envelope analysis of the airflow signal to improve polysomnographic assessment of sleep disordered breathing. Sleep 01 2014; 37:199–208.

[8] Springer DB, Tarassenko L, Clifford GD. Logistic regression-hsmm-based heart sound segmentation. IEEE Transactions on Biomedical Engineering April 2016; 63(4):822–832. ISSN 0018-9294.

[9] Hamilton P. Open source ecg analysis. In Computers in Cardiology, 2002. IEEE, 2002; 101–104.

[10] Ruder S. An overview of multi-task learning in deep neural networks. CoRR 2017;abs/1706.05098.

Address for correspondence:

gorogm@gmail.com