# Automated Sleep Stage Scoring from EEG data using Deep Learning techniques

**Nidhi Joshi**

*Supervisors*: Dr. Patrick Krauss, Dr. Achim Schilling, Prof. Andreas Maier

## Thesis Description

Sleep is commonly divided into 5 stages: 0 (Wake), 1 (REM, rapid eye movement), 2 (N1), 3 (N2), 4 (N3) on the basis of EEG data. The common practice for scoring sleep is that a somnologist or sleep lab nurse manually goes through the entire night and rates the sleep stage for each 30 second epoch of the polysomnographic signals according to the visual classification rules. This tedious and time-consuming process takes an experienced sleep scorer around 30-90 minutes for an 8-hour recording. As most of the sleep laboratories have multiple patients per night, the evaluation of recordings can therefore be considered an expensive economic factor. Furthermore, the inter-rater agreement rate of human scorers is only around 80% [1]. Thus, there arises a need to automate the process of sleep stage scoring.

Deep neural networks comprising of combination of Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) has worked well in this regard [1][2]. Here, exclusively the data of one EEG electrode placed centrally on the skull was used as the training data set for the neural network.

A neural network with eleven 1D convolutional layers and two bidirectional LSTM layers has resulted in an accuracy of 70% on our validation set. CNNs are used here to extract features from raw signal data. They typically serve the purpose of dimensionality reduction. Extracted features are then given as an input to LSTMs which are trained to deal with the temporal information of the data and thus are ideal to analyze time series data such as EEG signals. In this manner, the CNN-LSTM network learns features from the EEG data along with their sequence pattern to better predict the underlying sleep stage labels. Finally, the network predicts sleep stages (0-4) for the given channel data.

From this starting point, the described network will be used to address further interesting issues:

First, the trained network should be applied to the two other EEG channels (frontal, occipital) to quantify the impact of the electrode position on the validity of the determined sleep stage. Furthermore, there is some indication for a temporal propagation of the different sleep stages over the cerebral cortex, meaning that potentially different cortex areas are potentially in different sleep stages.

Second, the neural network can be used to evaluate the sleep data with a better temporal resolution. As described above, sleep stages are manually classified by the analysis of 30 second intervals. This time interval can be decreased (5s) to achieve a more fine-grained analysis.

Multidimensional Scaling (MDS) plots are used to view the internal representations of data of each class at every important layer of the neural network. In order to measure class separation, a metric called Generalized Discrimination Value (GDV) is calculated and plotted against the layers of the neural network [4].

Another important plot in the thesis is the 'hypno-density' plot, which measures the confidence votes of each class label (sleep stages). This visualization technique does not exclusively account for the sleep stage itself but provides a measure of the uncertainty. Pearson correlation is used to measure the correlation between any 2 channels of a patient's data.

The framework used for developing the algorithms is Keras with TensorFlow backend. Programming language is Python.

# References


[1] Simon Johannes Kern. Automatic Sleep Stage Classification using Convolutional Neural Networks with Long Short-Term Memory. Radbound University.

[2] Erik Bresch, Ulf Großekathöfer and Gary Garcia-Molina. Recurrent Deep Neural Networks for Real-Time Sleep Stage Classification From Single Channel EEG. Frontiers in Computational Neuroscience. 12:85. doi: 10.3389/fncom.2018.00085.

[3] Swihart, B. J., Punjabi, N. M., & Crainiceanu, C. M. (2015). Modeling sleep fragmentation in sleep hypnograms: An instance of fast, scalable discrete-state, discrete-time analyses. Computational statistics & data analysis, 89, 1–11. doi:10.1016/j.csda.2015.03.001.

[4] Schilling, Achim, et al. "How deep is deep enough?-Optimizing deep neural network architecture." arXiv preprint arXiv:1811.01753 (2018).