Statistical Modeling & Pattern Recognition

Project: *Eye State Detection & Epilepsy Recognition*

Professor: ***Vassilis Digalakis***
Lab Supervisors: ***Dr. Vassilis Diakoloukas & Dr. Vasillis Tsiaras***

Students: ***Thodoris Papathanasiou & Tontou Nikoletta***

Technical University of Crete - School of Electrical & Computer Engineering - May 2018

Pattern recognition in medicine

Electroencephalography(EEG): electrophysiological monitoring method to record electrical activity of the brain.

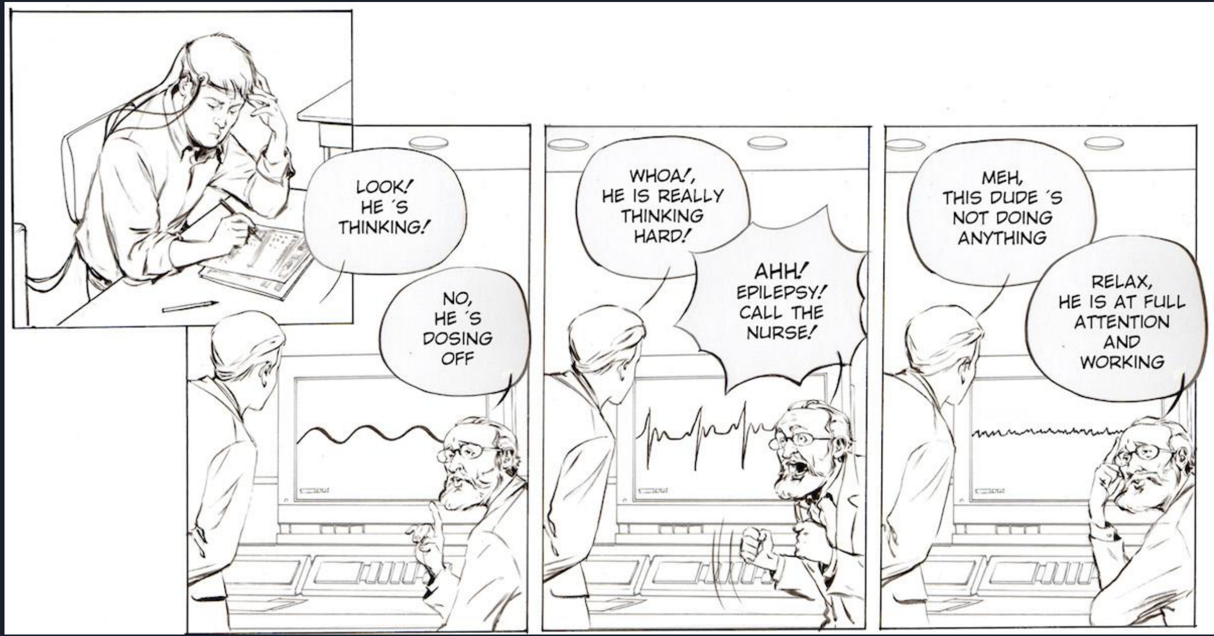
EEG is most often used to diagnose epilepsy, which causes abnormalities in EEG readings.

Advantage: very fast & real-time recordings

Pattern recognition in medicine

Disadvantage: difficult to interpret and results depend on doctor's skill.

Solution: Computer aided diagnosis!



EEG Eye State Detection

Dataset is from one continuous 117-second EEG measurement. Consists of 14 EEG values and a value indicating the eye state.
The eye state was detected via a camera during the EEG measurement and added later manually to the file after analysing the video frames. '1' indicates the eye-closed and '0' the eye-open state.

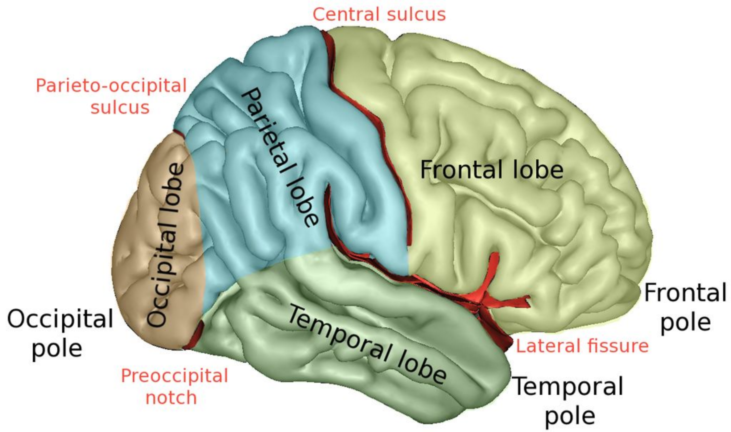
Number of Attributes: 15
Number of Instances: 14980

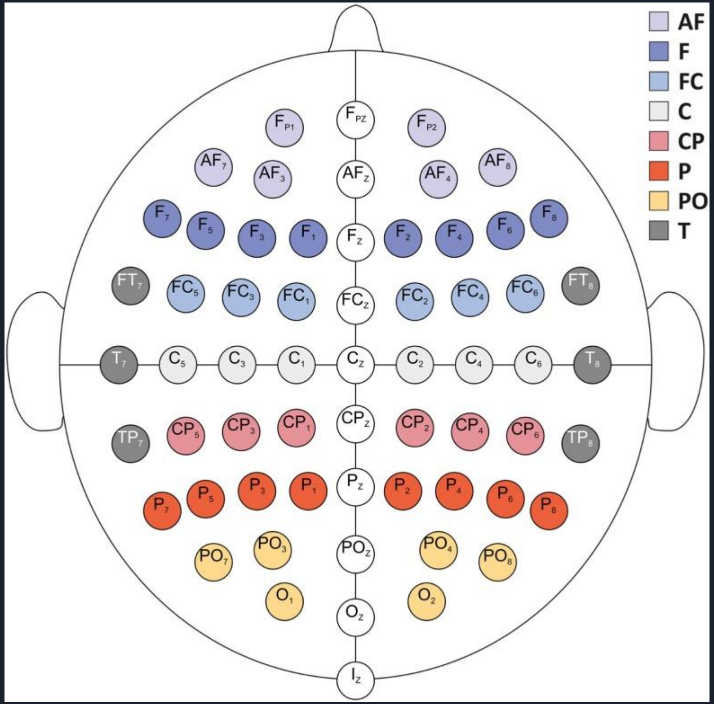
Electrode Placement

F = Frontal
P = Parietal
T = Temporal
O = Occipital
C = Central

A = Auxiliary

Odd# = Left
Even# = Right





Description of Features
#Matlab #Classification Learner

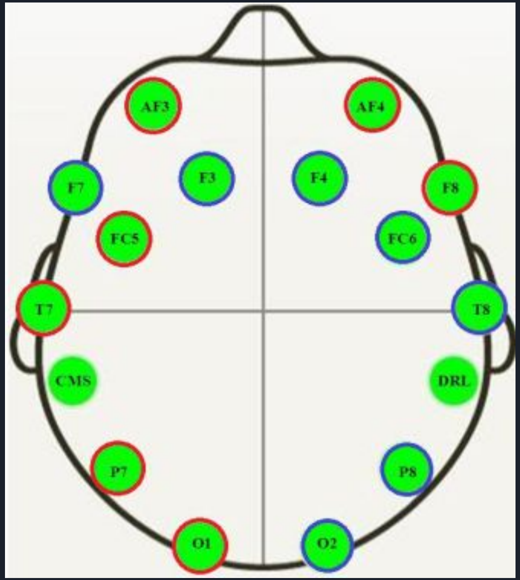
- Each electrode signal: *all the 14 EEG values*

- Electrodes into groups (all, dataLeft, dataRight):

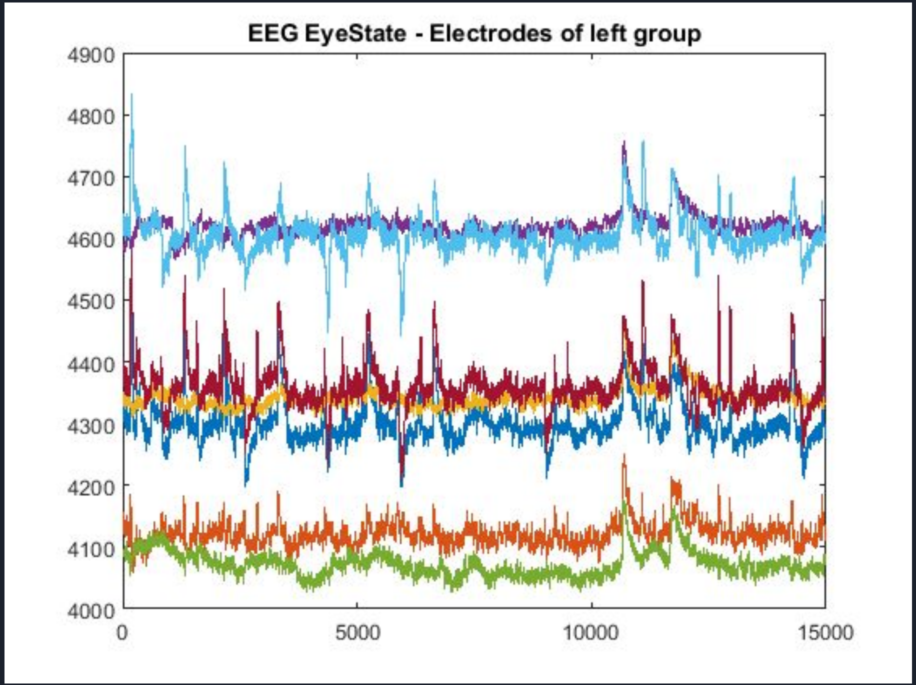
    - **Skewness:** *measure of asymmetry of the probability distribution of a real-valued random variable about its mean.*

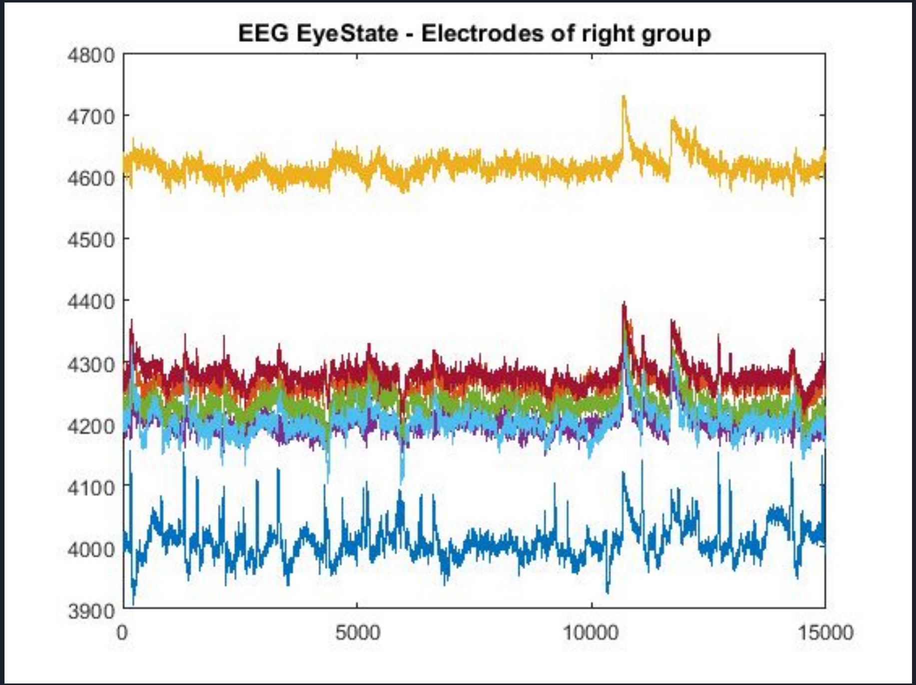
    - **Kurtosis:** *measure of the "tailedness" of the probability distribution of a real-valued random variable.*

    - Variance: *the expectation of the squared deviation of a random variable from its mean.*



Data left & right electrodes’ signals





Classification Experiments Using Classification Learner Algorithms

- Quad Discriminant → 78.7%
- Complex Tree (all data) → 77.8%
- Coarse Gaussian SVM (all data) → 71.1%
- Linear Discriminant → 64.1%
- Logistic Regression → 64.1%

Classification Experiments Using Classification Learner Algorithms

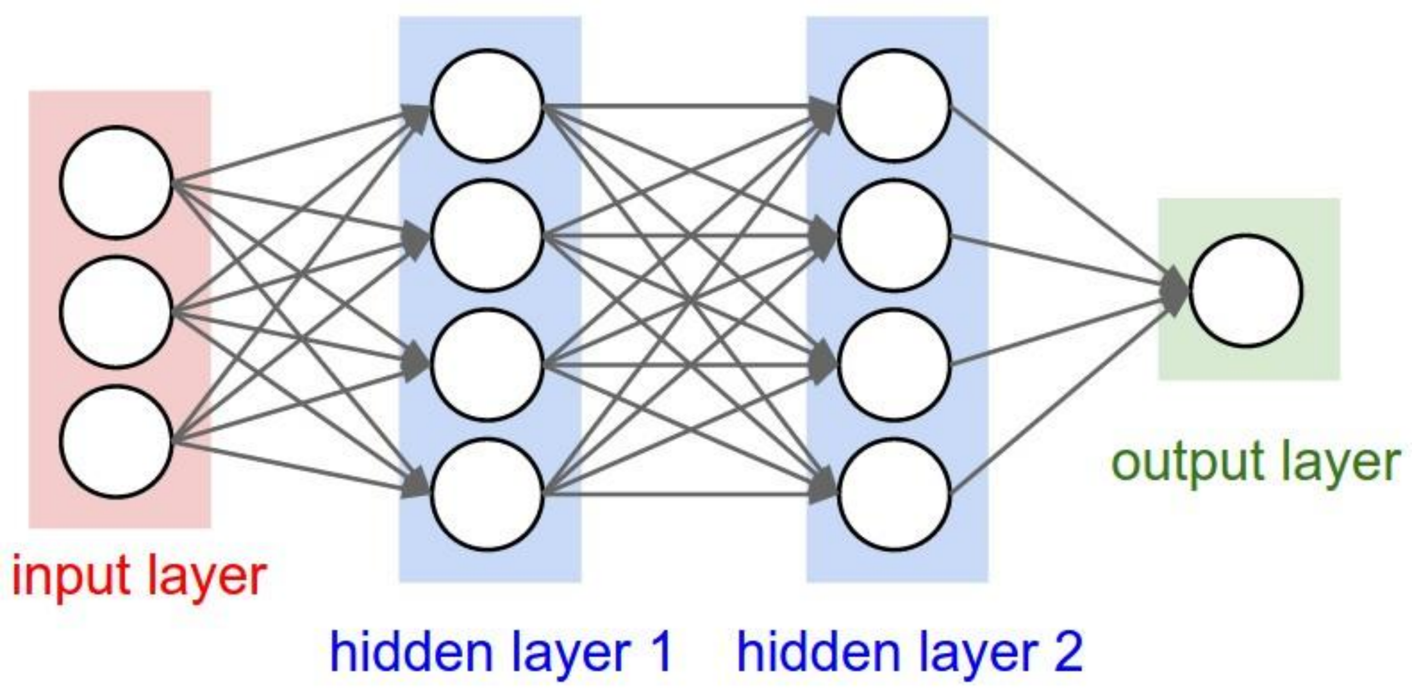
- Cubic SVM:
  - All data → **94.5%**

- Fine Gaussian SVM:
  - All data → **97.6%** | All data (with PCA) → 91.2%
  - Left data → 89.5%
  - Right data → 82.5%
  - Variance - Skewness -Kurtosis (v-s-k)
    - All data → 97.7% (with v-s-k)
    - Left data → 92.5% (with v-s-k)
    - Right data → 92.5% (with v-s-k)

Classification Experiments Using Classification Learner Algorithms

- Fine KNN:
  - All data → 97.3% | All data (with PCA) → 89.8%
  - Left data → 88.6%
  - Right data → 79.4%
  - Variance - Skewness -Kurtosis (v-s-k)
    - All data → **97.7% (with v-s-k)** | 97.5% (with v) | 97.5% (with s) | 97.4% (with k)
      → 97.3% (with v-s-k of right data) | **97.7% (with v-s-k of left data)**
    - Left data → 77.9% (with v-s-k) | 76.6% (with v) | 75.7% (with s) | 75% (with k)
      → 76.7% (with v-s-k of right data) | 75.7% (with v-s-k of left data)
    - Right data → 71.3% (with v-s-k) | 67.6% (with v) | 69.1% (with s) | 71% (with k)
      → 64.4% (with v-s-k of right data) | 75.6% (with v-s-k of left data)
  - Row Varience of each electrode
    - Data bigger than mean=638.4813 (with v-s-k) [7/17 features] → 85.9%
    - Data bigger than 400 (with v-s-k) [12/17 features] → 96%
    - Data bigger than 390 (with v-s-k) [13/17 features] → 96.9%

Neural Networks Classification



Data manipulation and preprocessing

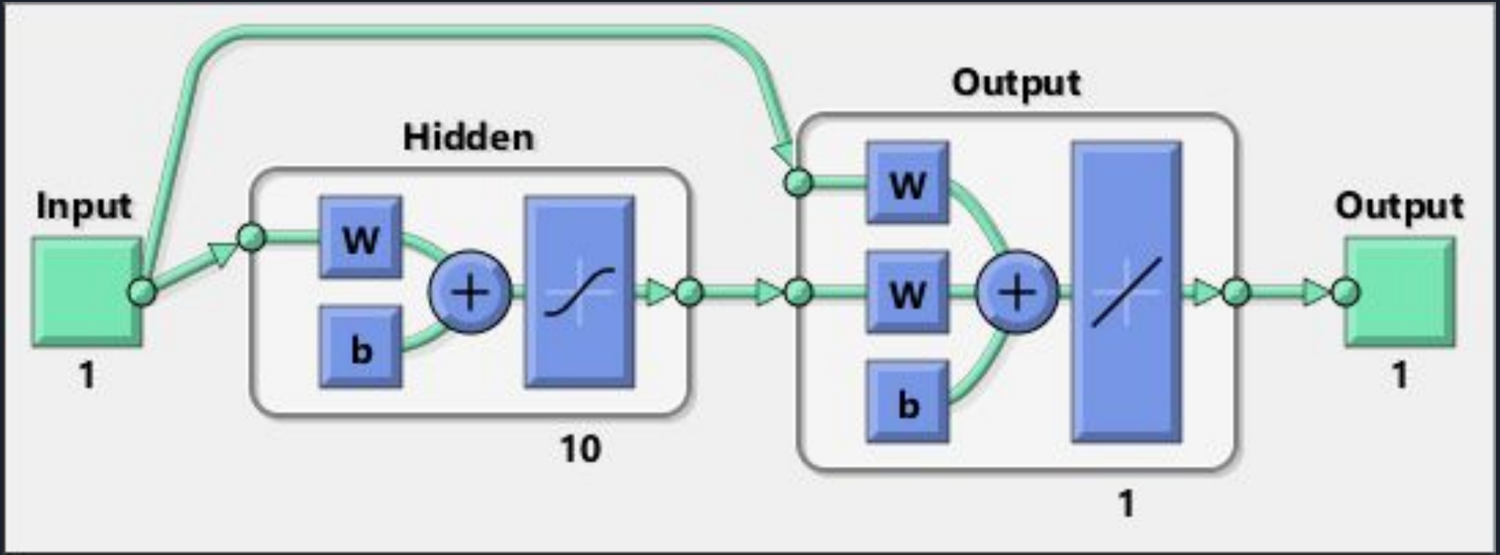
- Outliers have been removed through matlab script.
- Data is then read through a csv file and is loaded to two lists, data and labels.
- After that, we normalize the data list to 0, 1 based on simple max min normalization
- Finally we shuffle the data and the labels and we divide the set into training, validation and test sub-sets (80%-20%-64).

Implemented Architectures

Searching for the most appropriate neural network type, we implemented 3 different architectures.

- ❑ **Multilayer Perceptron feed forward neural network**: 2 fully connected hidden layers
- ❑ **Long short-term memory recurrent neural network**: 1 lstm unit, non-shuffled data
- ❑ **Cascade forward neural network**: 2 fully connected hidden layers and input-output connection

Cascade forward neural network



Experimental Results

- MLP: 200 training epochs, 64 batch size, 0,001 learning rate, Adam optimizer
  - Avg. test accuracy: 0,577
  - With log(data) avg. test accuracy: 0,524

- LSTM-RNN: 100 training epochs, 64 batch size, 0,001 learning rate, Adam optimizer
  - Avg. test accuracy: 0,557

Experimental Results cont.

- CFNN: 200 training epochs, 64 batch size, 0,001 learning rate, Adam optimizer
  - 2 fully connected (dense) layers. 1st layer ReLU activation, 2nd layer dropout.
  - Input is also cascaded to the output tensor via appropriate weights.
  - Avg. test accuracy with strict over-fitting control: 0,719
  - Avg. test accuracy with loose over-fitting control: 0,794

Therefore CFNN is the only option that shows potential for an acceptable test accuracy.

Future Work

- Optimization of Fine Gaussian KNN to maybe achieve scores close to 99%
- Our current CNN implementation for epilepsy dataset achieves ~95% accuracy. Manipulate eye-state dataset to be compatible with the aforementioned CNN or a modified version.
- Parameter optimization for CFNN as to achieve accuracy comparable to Fine Gaussian KNN