ELECTROENCEPHALOGRAM-BASED BRAIN COMPUTER INTERFACE

CENA MILLER
UNIVERSITY OF WYOMING – ELECTRICAL ENGINEERING
ACKNOWLEDGMENTS: DR. SURESH MUKNAHALLIPATNA

PROJECT OVERVIEW

Analyzed large data sets of brain activity measurements for patterns that indicated which direction a person was thinking about moving

↓

Controlled the motion of a ball in a simple program, by looking for those patterns in new data, as it was being measured

BACKGROUND

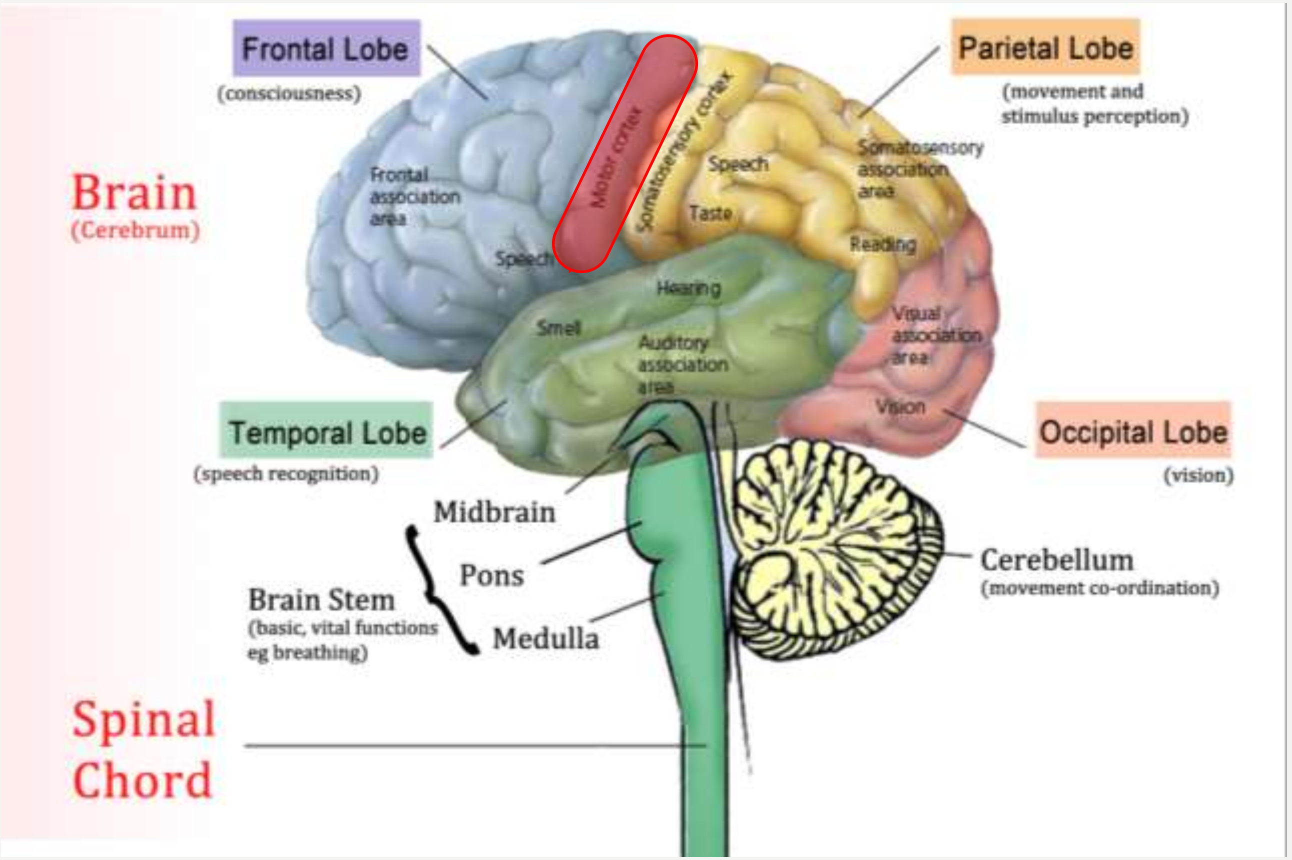


Image of Central and Peripheral Nervous system from : http://www.biologyreference.com/Oc-Ph/Peripheral-Nervous-System.html

Motor Cortex:

Controls voluntary movement

PROBLEM STATEMENT

The connection between the motor cortex and peripheral muscles may be damaged.

Nearly 8 million Americans live with impaired motor function, due to paralysis or the loss of limbs.

Brain-Computer Interfaces (BCIs) use activity in the brain to control hardware or software

Medical Applications

If we can process and classify activity from the motor cortex in real-time, we can build intuitive hardware that uses those signals.

(2) BCI developed at University of Houston, engineering research led by Dr. Jose Contreas-Vidal

ELECTROENCEPHALOGRAPHY (EEG)

GOAL: IMPLEMENT BCI

- Provide raw EEG data from headset to data processing software
- Classify three types of brain activity generated by motor activity
- Create simple GUI to demonstrate my interface

Hardware

Emotiv Epoc+ EEG Headset ($799)

- Sampling Rate: 128 Hz
- 14 Channels of EEG Data
- Wireless Bluetooth Connection with 2-3 meter range
- 6 hour battery life

Software

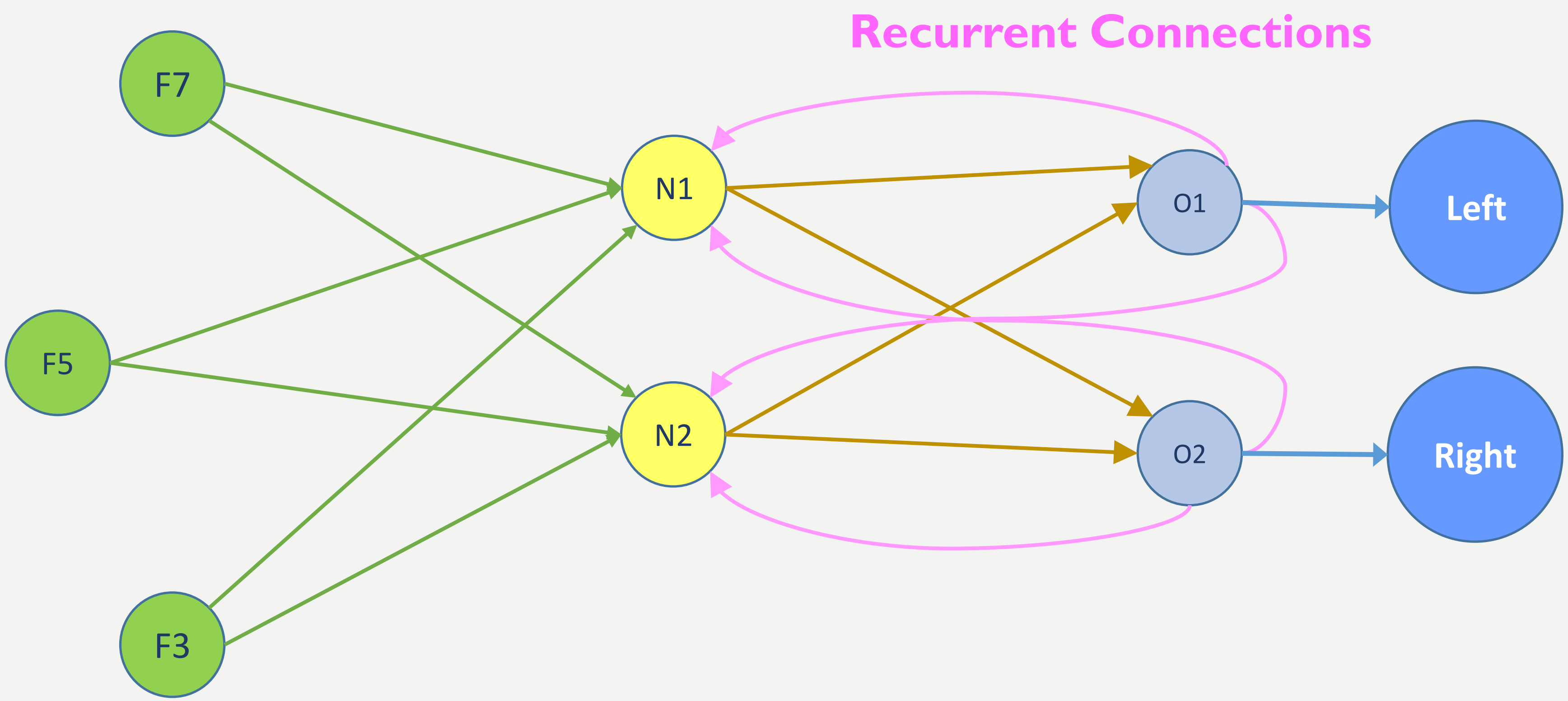
- Network designed and trained in MATLAB 2016b Neural Net Toolbox ($39)
- Real-time data processing and GUI implemented in Python 2.7

NEURAL NETWORK ARCHITECTURE TO CLASSIFY MOTOR IMAGERY

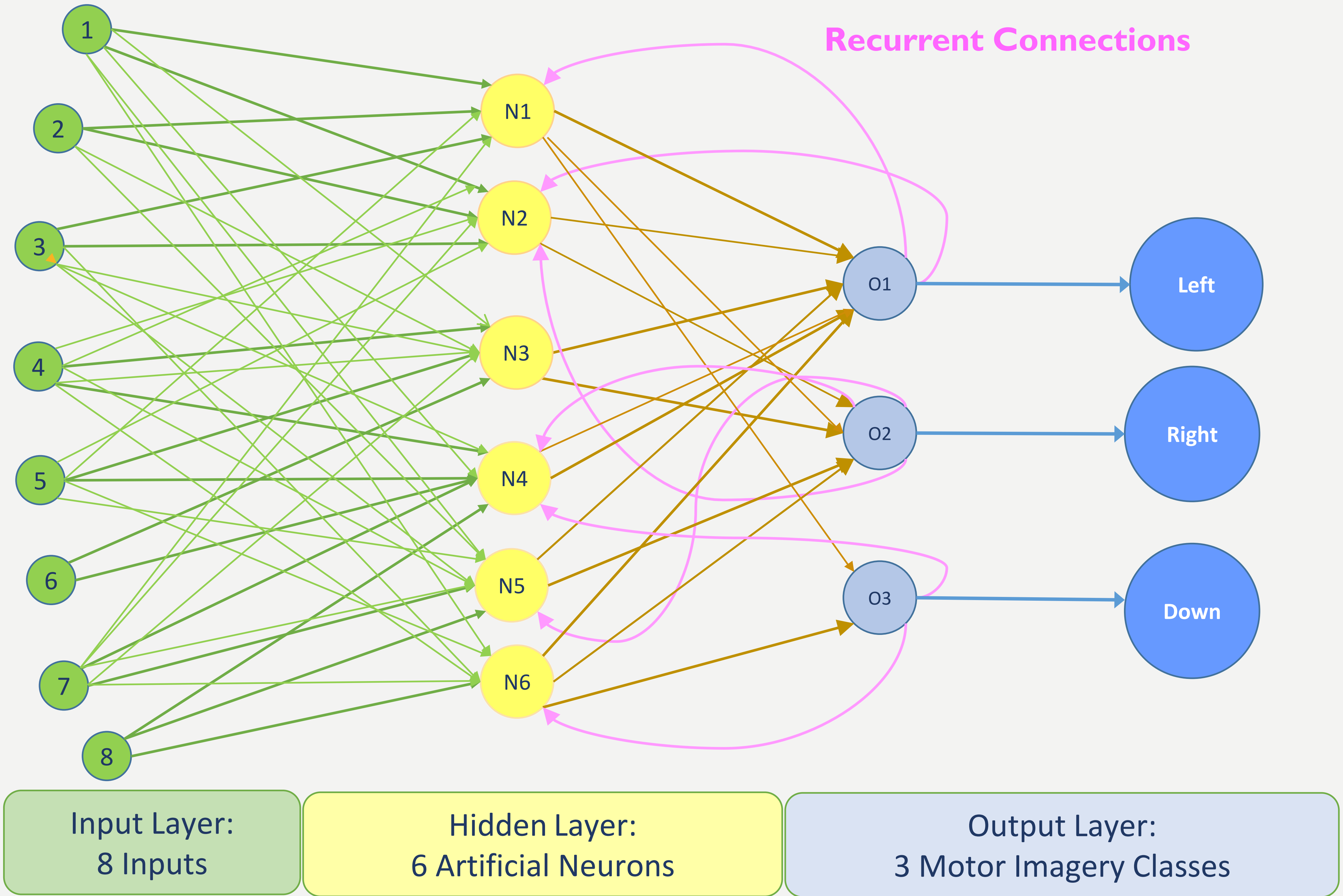
TRAINING DATA

- Anonymous data sets from seven individuals, provided by Berlin BCI group
- Three classes of motor imagery:
    - Left (*Left Hand*)
    - Right (*Right Hand*)
    - Down (*Feet*)
- 59 sensor locations
- Eight data channels were selected for analysis
- Target output vectors were generated for each time sample, based on provided cue-timing data

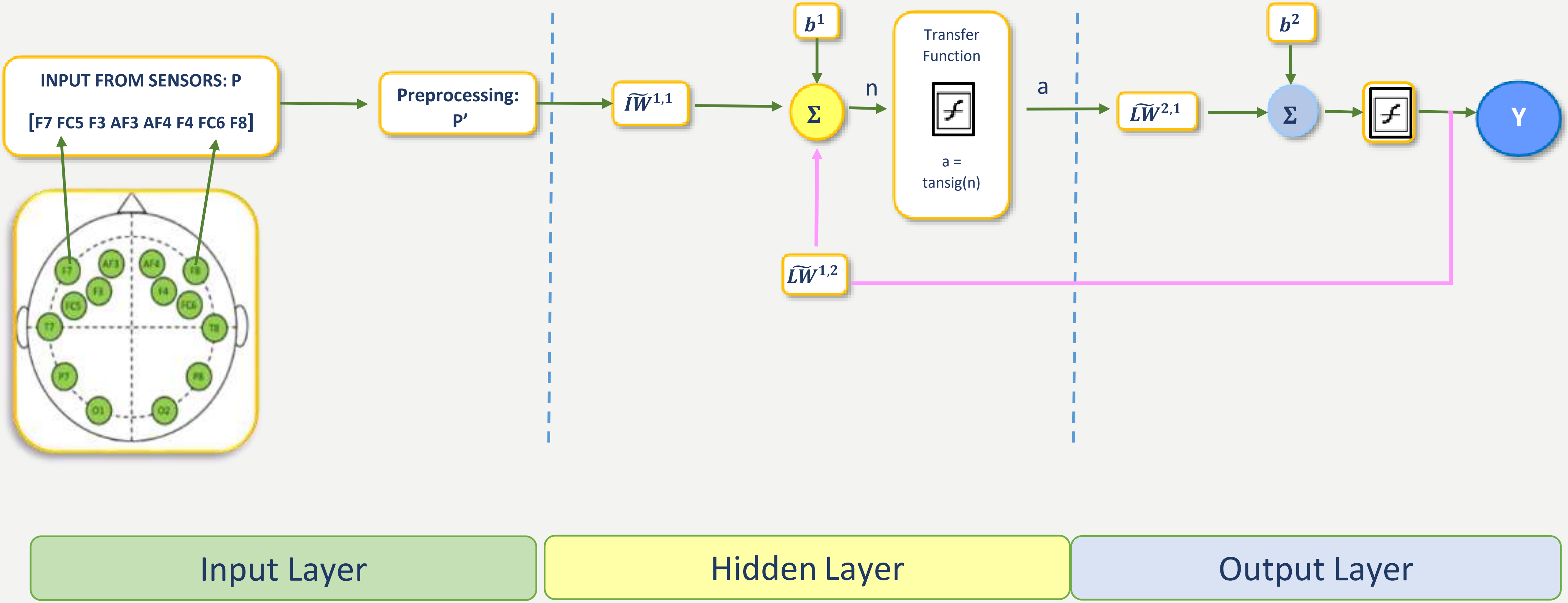
EXAMPLE RECURRENT NEURAL NETWORK



Input Layer: 3 inputs	Hidden Layer: 2 Artificial Neurons	Output Layer: 2 Motor Imagery Classes


Recurrent Connections
1
2
3
4
5
6
7
8
N1
N2
N3
N4
N5
N6
O1
O2
O3
Left
Right
Down
Input Layer:
8 Inputs
Hidden Layer:
6 Artificial Neurons
Output Layer:
3 Motor Imagery Classes

SINGLE PATH THROUGH NETWORK



TRAINING ALGORITHM

Levenberg-Marquardt Algorithm

- Iterative process
- Calculate the incremental changes in weight and bias variables (ΔX), depending on the Mean Square Error (MSE) that is calculated for each training input sample

$$MSE = \frac{1}{3}\sum_{i=1}^{3}(Output_i - Target_i)^2$$

TRAINING ALGORITHM

Levenberg-Marquardt Algorithm

- Next, the Jacobian matrices of each bias and weight vector (*jX*) are calculated as the 1st derivative of the MSE with respect to each X

- To approach a local minimum in error performance, X is adjusted towards the direction of the negative of the error gradient (*je*)

$$je = jX^T * MSE$$

TRAINING ALGORITHM

Levenberg-Marquardt Algorithm

- To increase the speed of convergence, the 2nd derivative of MSE with respect to X is approximated as

$$jj = jX^T * jX$$

- Finally, ΔX is calculated

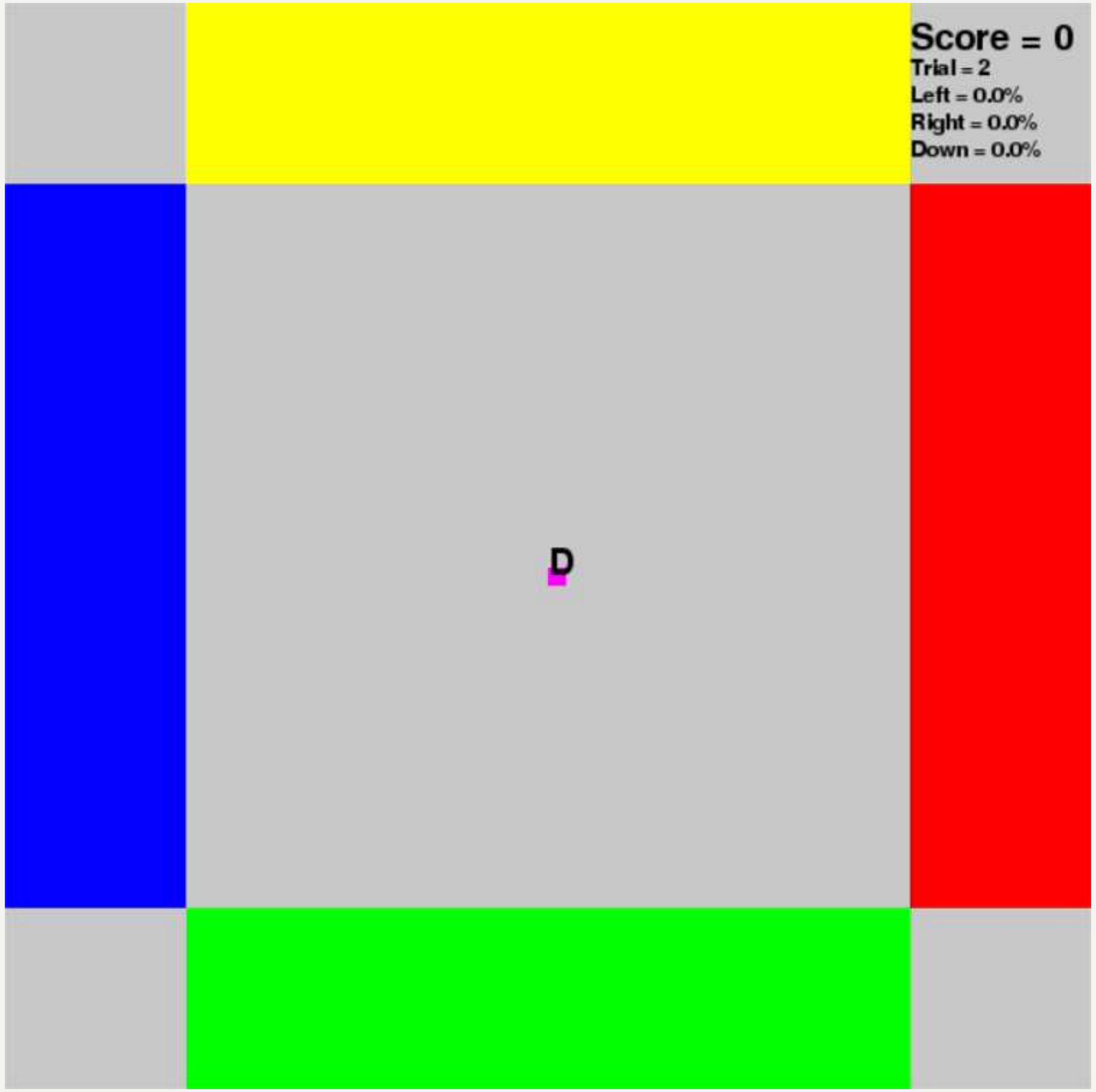
$$\Delta X = [jj + \mu I]^{-1} * je$$

Where μ is iteratively increased until the error is decreased

NETWORK TRAINING RESULTS

TRAINING RESULTS			
	Samples	**MSE**	**R**
Training	2,049,184	5.46e-4	.997
Testing	439,111	5.73e-4	.973

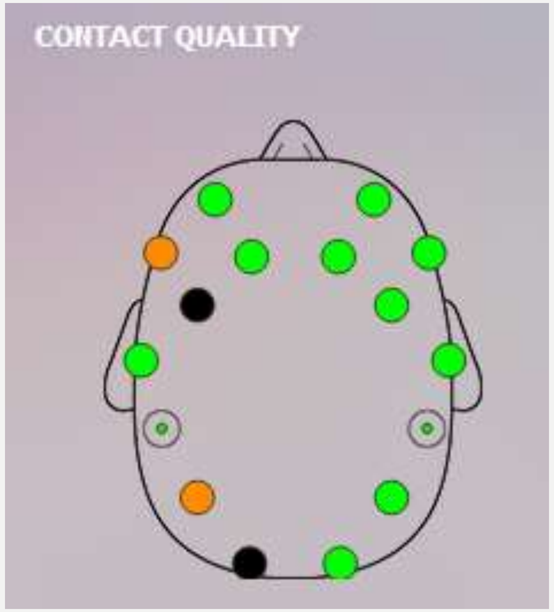
GRAPHICAL USER INTERFACE



GUI RESULTS			
Trials	**Left Score**	**Right Score**	**Down Score**
500	46.6%	33.6%	43.1%

Challenges

- Sensor quality & placement
- Network Overfitting



RESULTS

Training the network with seven anonymous data sets did not provide a reliable and generalizable classifier for the Brain-Computer Interface.

Further training with data recorded by the Epoc+ headset may improve performance.