Master in Artificial Intelligence and Robotics

SAPIENZA
UNIVERSITÀ DI ROMA

Classification Test of EEG signals Using Neucube

Francesca Palermo
ID: 1401074

Giulio Marano
ID: 1401074

Introduction

In this project we are going to thoroughly describe the working principle of the software Neucube through the classification of an EEG Database Dataset.

The report is divided as follows:

- The problem of classification and a brief description of the Spike-Timing-Dependent Plasticity are taken upon.
- The dataset used is illustrated in detail.
- The software NeuCube is discussed together with a general analysis of its Interface
- The project is deeply described step by step with the confrontation of the same dataset with different channels (64,16 and 6) and some trials made to optimize the results.

Spike-Timing-Dependent Plasticity

STDP is a temporally asymmetric form of Hebbian learning induced by tight temporal correlations between the spikes of pre- and postsynaptic neurons.

STDP allows to automatically balance synaptic weights in such a way to obtain an irregular post-synaptic firing but sensible to the timing of the pre-synaptic spikes.

The modified synapsis compete for the control of the post-synaptic action potentials. Inputs which are able to fire neurons post-synaptic with a reduced delay are able to compete with greater success, while synapsis with larger delay or with less effective inputs, become weaker.

$$\Delta\omega_j = \sum_{f=1}^{N} \sum_{n=1}^{N} W({t_i}^n - {t_j}^f) \qquad W(x) = A_+ \exp\left(-\frac{x}{\tau_+}\right) \quad \text{for x} > 0$$

where

- $\Delta\omega_j$ is the total weight change
- ${t_j}^f$ is the presynaptic spike arrival times at synapse j with f the number of presynaptic spike
- ${t_i}^n$ is the firing time of the postsynaptic neuron

Classification Problem

The goal in classification is to take an input vector $\boldsymbol{x}$ and to assign it to one of K discrete classes C_k where $K = 1, \ldots, K$

The input space is thereby divided into decision regions whose boundaries are called decision boundaries.

A classifier is a systematic approach to building classification models from an input Data set.

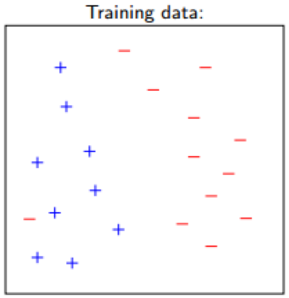
Each technique employs:

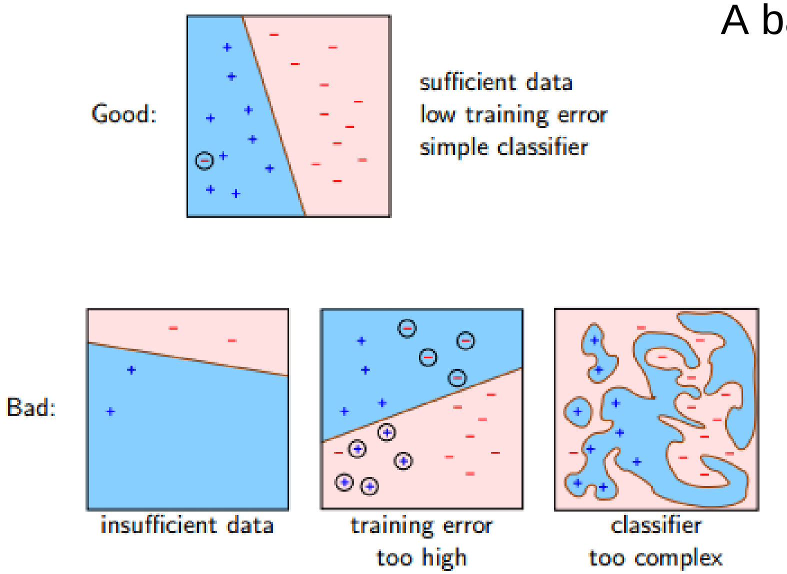
- Learning algorithm
- Training set
- Test set
- Confusion matrix
- Performance metric accuracy

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

For good test peformance, we need:

- enough training examples
- good performance on training set
- classifier that is not too “complex” (“Occam’s razor”)

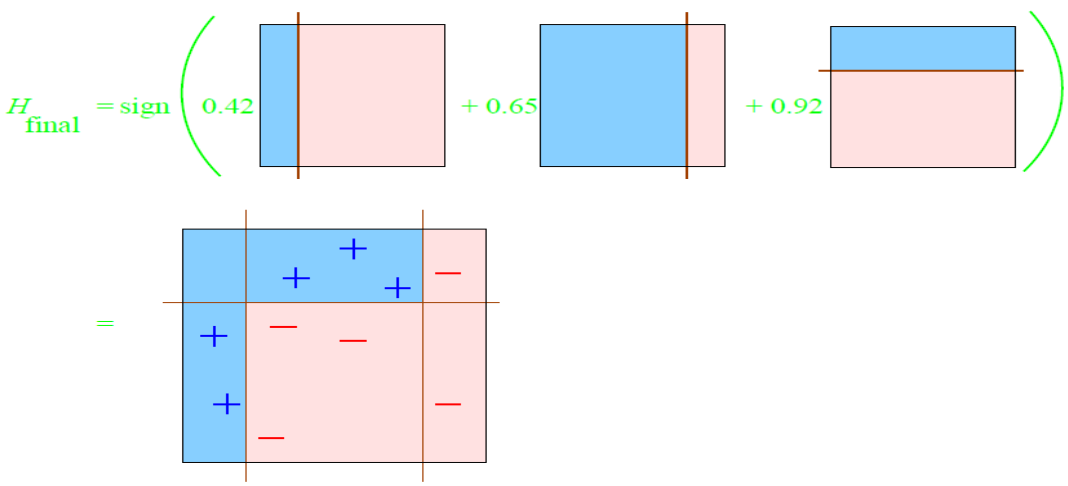




A bad classifier can be turn around by more simply round stages

Boost approach

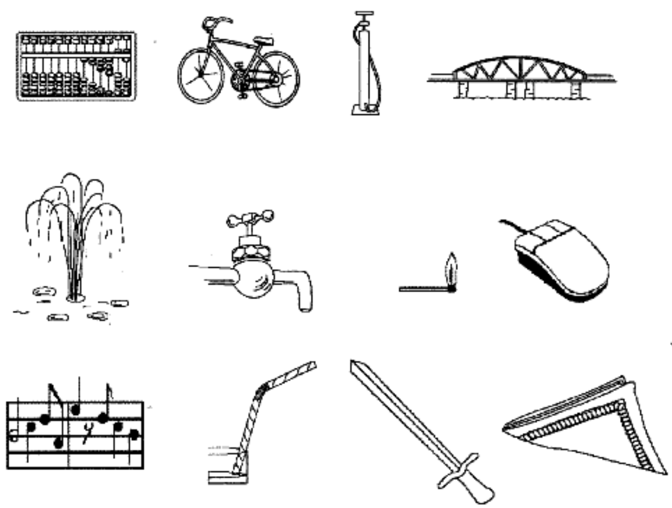
Boosting = general method of converting rough rules of thumb into highly accurate prediction rule



Dataset Description

Two group of subjects:

- Alcoholic
- Control



And three stimulus classes:

- S_1, one single stimulus
- S_1 and S_2, $S_1 \neq S_2$
- S_1 and S_2, $S_1 = S_2$

There were 122 subjects and each subject completed 30 trials. The electrode positions were located at standard sites, following Standard Electrode Position Nomenclature

There were three classes:

- Small Data Set (only 2 subjects)
- Large Data Set (20 subjects) [The one used in the project]
- Full Data Set (122 subjects)

Since the datasets are not in a format accepted by NeuCube, we have coded a Matlab's script to automatically convert it. In the accepted format, data have a row for each trial divided by commas for each channel

Attribute Information:

Each trial is stored in its own file and will appear in the following format.

co2a0000364.rd
120 trials, 64 chans, 416 samples 368 post_stim samples
3.906000 msecs uV
S1 obj , trial 0
FP1 chan 0
0 FP1 0 -8.921
0 FP1 1 -8.433
0 FP1 2 -2.574
0 FP1 3 5.239
0 FP1 4 11.587
0 FP1 5 14.028
...

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	-3550	-4476	-4313	-1200	-7507	-8921	-7741	-4100	24017	-2319	1780	-488	2391	-2574	-2716	-2625		
2	-5015	-3499	-2848	1241	-5066	-7456	-6276	-4588	-21881	610	1292	1465	1414	-2085	-3204	-2625		
3	-5503	-1058	-407	2706	-671	-4527	-2370	-6053	-3815	2075	-661	2441	-51	-1597	-3693	-2625		
4	-3550	1383	1058	3682	1770	-2574	1048	-7517	-5280	-1343	-3591	2930	-1027	-1597	-4669	-3113		
5	-621	3337	1546	3682	1282	-3062	71	-8982	580	-1343	-5544	2930	-1516	-1597	-5646	-3113		
6	1821	2848	570	3194	-2136	-4527	-4812	-9959	1068	-1343	-6032	2441	-1516	-1597	-6622	-3601		
7	2309	2360	81	2218	-4578	-3062	-10671	-10447	5463	-854	-4079	2441	-1027	-1109	-6134	-3113		
8	844	2848	570	1241	-3601	3286	-12136	-9471	6439	-2808	-2126	2441	-539	-621	-4669	-2136		
9	844	4801	1546	753	793	12075	-6765	-5564	-6256	3052	-173	2441	-539	-132	-2228	-1160		
10	3286	7731	2523	264	6165	18422	1536	295	-397	2075	804	1465	-539	-132	-763	-671		
11	7680	8708	3011	264	8606	16469	7395	7619	92	-2808	315	0	-1027	-1597	-763	-2136		
12	10122	7731	2523	753	7141	6704	7395	11525	-1373	-854	-173	-1465	-1027	-4527	-4181	-4578		
13	9145	3825	1546	1729	4211	-4038	1536	10061	580	122	315	-2930	-1027	-7456	-7599	-7019		
14	5239	-570	1058	2706	2258	-9410	-4323	3713	3510	-4272	-173	-4395	-2004	-9410	-10040	-8972		
15	844	-3011	2523	4171	3235	-7456	-6276	-3611	3021	-4761	-2614	-4883	-3469	-10874	-11505	-9460		
16	-621	-2035	4476	4659	5676	-2574	-3835	-7029	1556	-3784	-6032	-6348	-4934	-10874	-11505	-10437		
17	2309	1383	6429	5147	8118	-132	559	-6053	1556	-1831	-8474	-7324	-6399	-10874	-11993	-10925		
18	6215	5290	6917	4171	8606	-3550	3489	-2146	92	-366	-8962	-8301	-5910	-10386	-11505	-10925		
19	7192	5778	5452	2706	7629	-9898	2513	295	3998	-4272	-6032	-7812	-3957	-9898	-11017	-10437		
20	4262	3825	3011	1729	5676	-14781	-905	-1170	-2838	610	-2126	-6836	-2004	-8921	-9552	-8484		
21	-1109	-570	570	264	4211	-14781	-4323	-4588	-2350	1587	315	-5371	-2004	-7945	-8575	-7019		
22	-5503	-4476	-895	-224	2747	-10874	-6276	-7029	-1862	-2808	-661	-3906	-2981	-7456	-7599	-7019		
23	-6480	-5452	-407	-224	1282	-5992	-6276	-5564	2533	-5737	-3591	-3906	-4445	-7945	-8087	-7507		
24	-4038	-3988	570	-712	-183	-4038	-5300	-1170	1068	-4272	-7009	-3906	-4445	-7945	-9552	-8484		
25	-132	-2035	2035	-224	-2136	-5015	-4323	2736	3998	-4272	-7497	-3418	-2981	-7456	-10040	-8972		
26	1821	-2035	2035	-712	-2625	-7456	-4323	3225	-397	-2808	-5544	-2441	-1516	-6968	-10040	-8484		
27	1821	-3499	1546	-712	-1160	-8433	-4323	1272	23041	-8667	-2126	-1953	-1027	-7456	-9552	-7507		
28	356	-5941	1058	-1200	1770	-6968	-2858	-682	-22858	-854	-661	-2930	-2004	-8433	-8087	-7996		
29	-132	-6429	1546	-2177	4211	-4038	-417	-1170	-9186	1587	-1149	-5371	-4445	-10386	-8087	-9460		
30	1821	-3011	3011	-1689	6165	-2574	2513	295	-3326	-3296	-3103	-8789	-6399	-12339	-9552	-10925		
31	4262	3825	5941	264	5676	-4038	3977	1760	-2350	-3784	-5544	-11719	-7375	-13804	-11993	-12878		

16x256

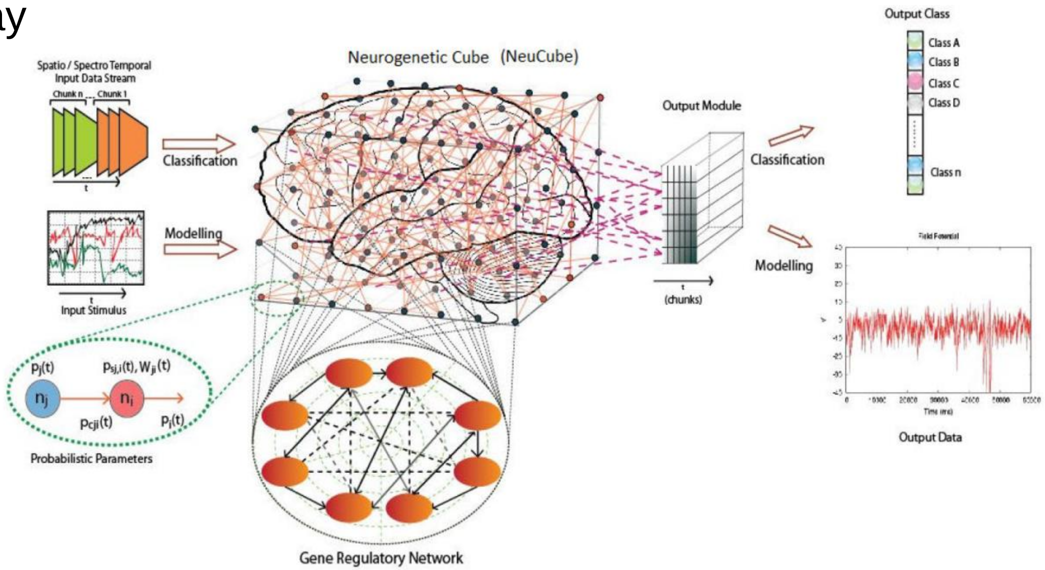
Seeing that the original number of channels (64) created problems in reading the results, we tried to reduce it and to study the different behaviour of the classifier with 64, 16 and 6 channels.

Neucube

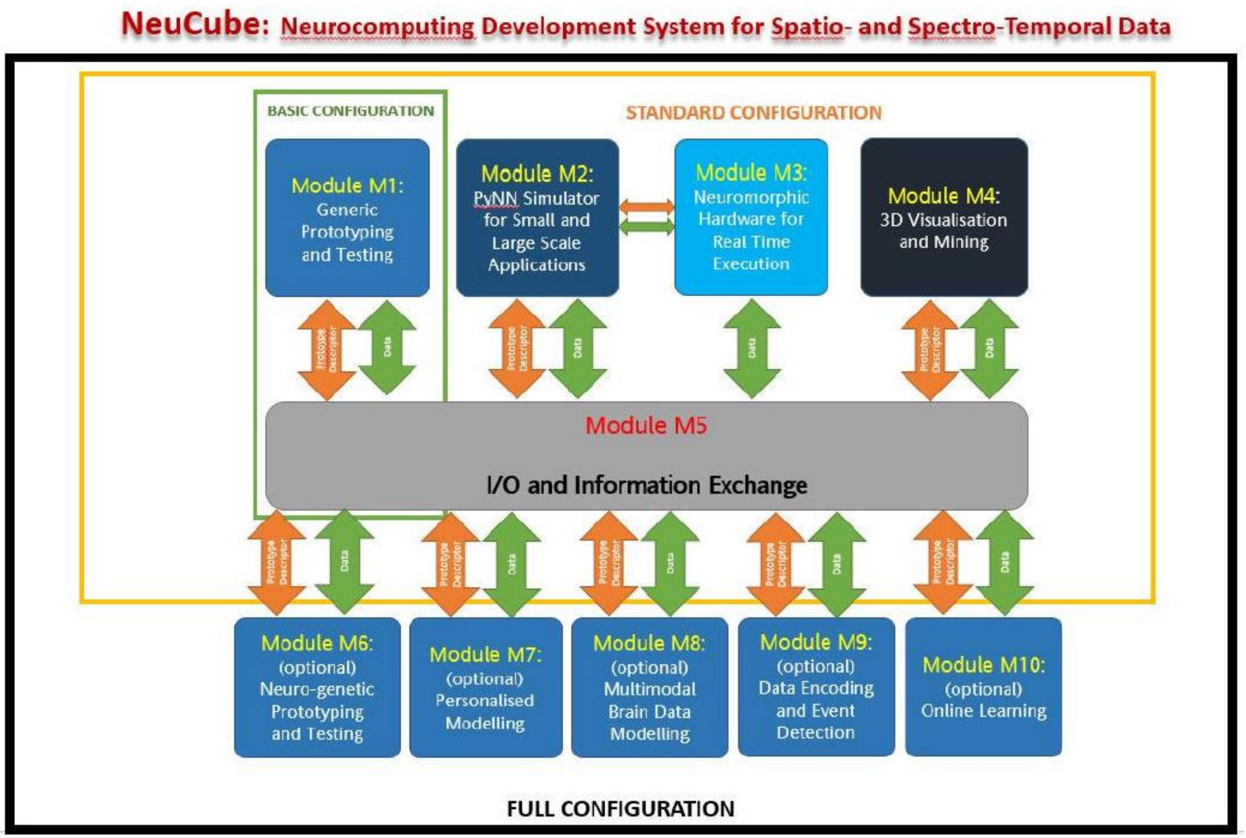
NeuCube is a software development environment for SNN prototype systems. It facilitates the design and the implementation of efficient solutions to problems through precise selection and testing of most suitable methods and parameters for an STDM (Spatio-Temporal Data Machine)

An STDM has three parts:

- an input part to encode input data into spiking sequences
- an SNNcube that learns the input data in an unsupervised mode to capture spatio-temporal patterns
- an evolving output part for classification tasks that is trained in an incremental, adaptive way

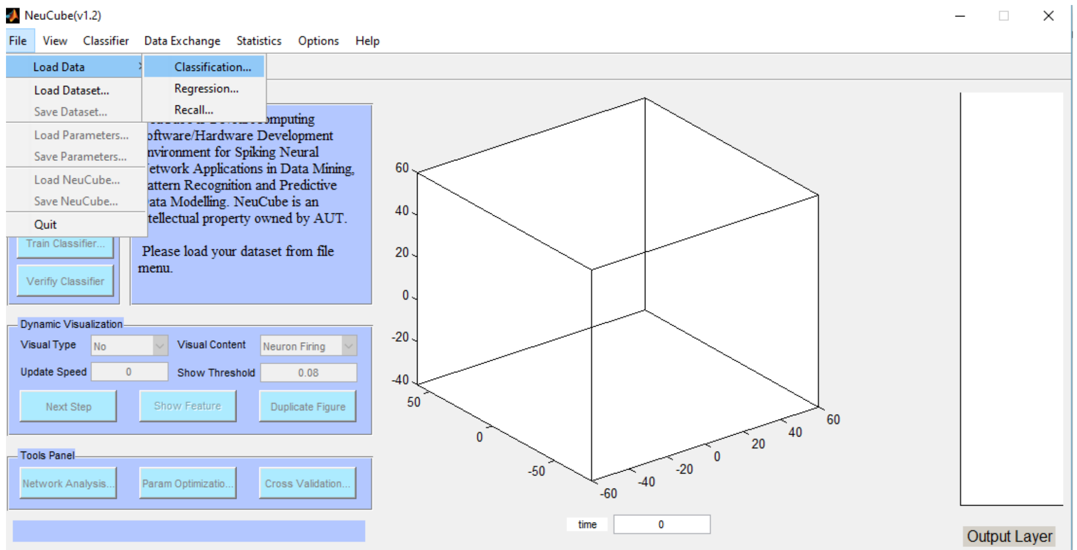


The previous architecture was developed further as a multi-modular software/hardware development system applications.



Analysis of the project
1. Loading a Dataset

Data can be loaded from the menu by clicking file→load dataset→classification



These images show how the metadata of the project dataset with respectively 64, 16 and 6 channels is displayed in the information panel after the data is successfully loaded

Dataset Information:
sample number: 120
feature number: 64
time length: 256
class number: 2

Task Type: Classification

Dataset Information:
sample number: 120
feature number: 16
time length: 256
class number: 2

Task Type: Classification

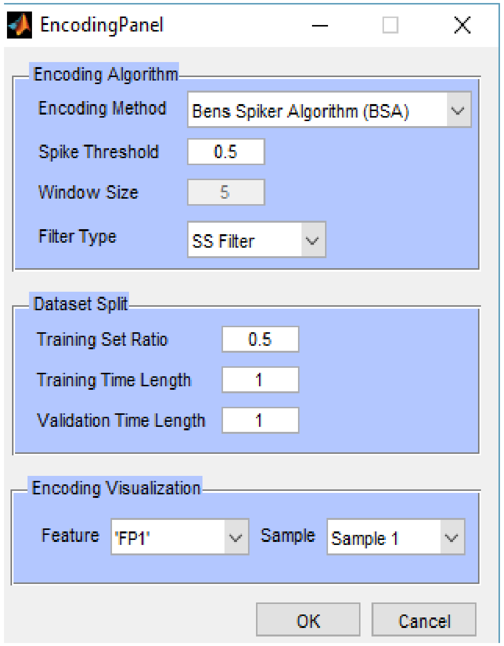
Dataset Information:
sample number: 120
feature number: 6
time length: 256
class number: 2

Task Type: Classification

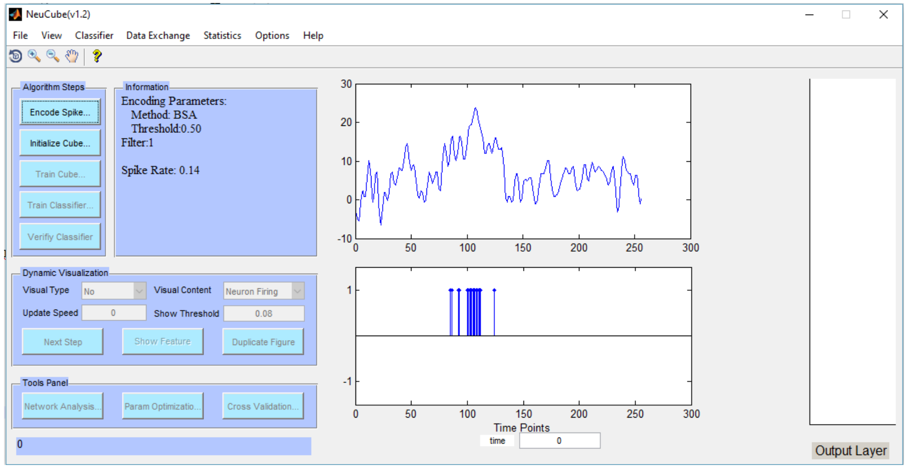
2. Data Encoding

The next step is to encode the real-value input data into trains of spikes.

Clicking the “*Encode Spike*” button generates a new UI panel



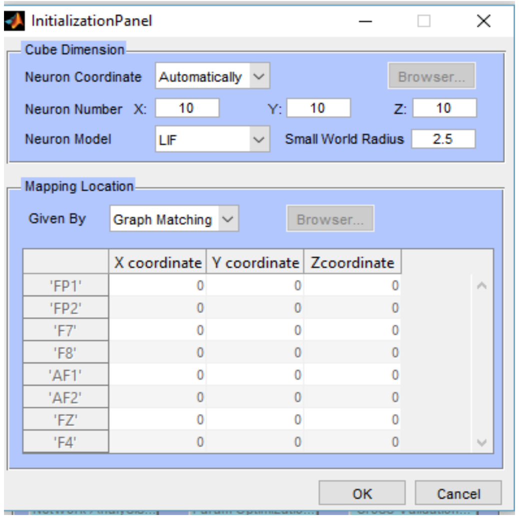
The graph on top shows the raw input data for the chosen sample and feature while the bottom one shows the positive and negative spike trains generated from the raw data.
(64 channels for brevity)



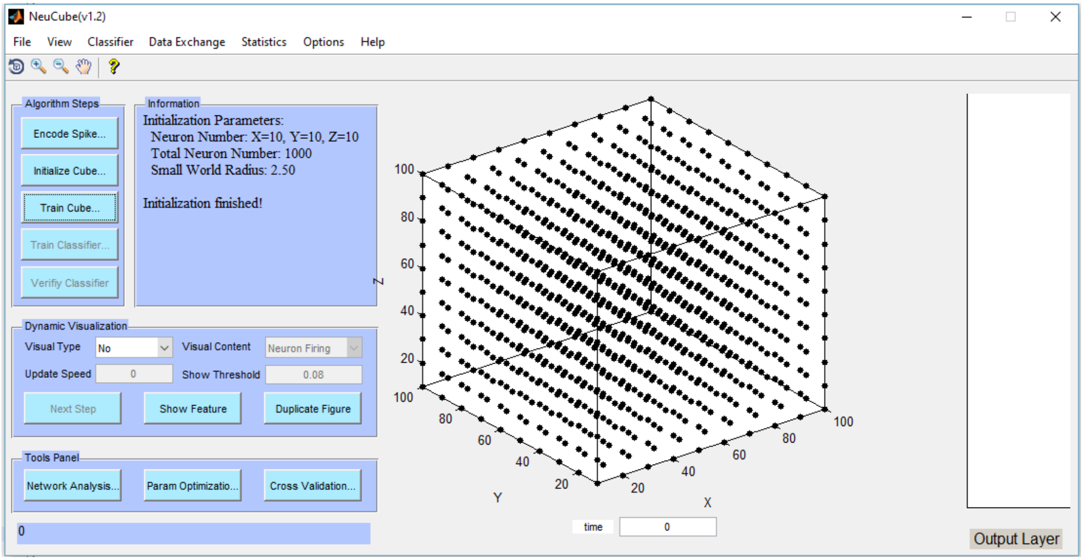
3. Cube Inizialization

The next step is to initialize the Cube by clicking the “*Initialize Cube”* button.

The above subpanel is used to configure the properties of the neurons in the cube. The below subpanel shows the coordinates of the input neurons.



Once the initialization is finished, the “*3D visualization*” panel shows the initialized cube

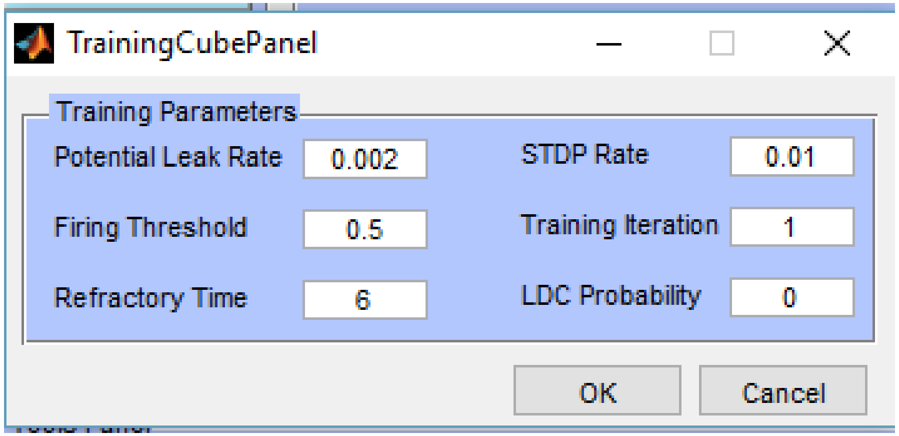


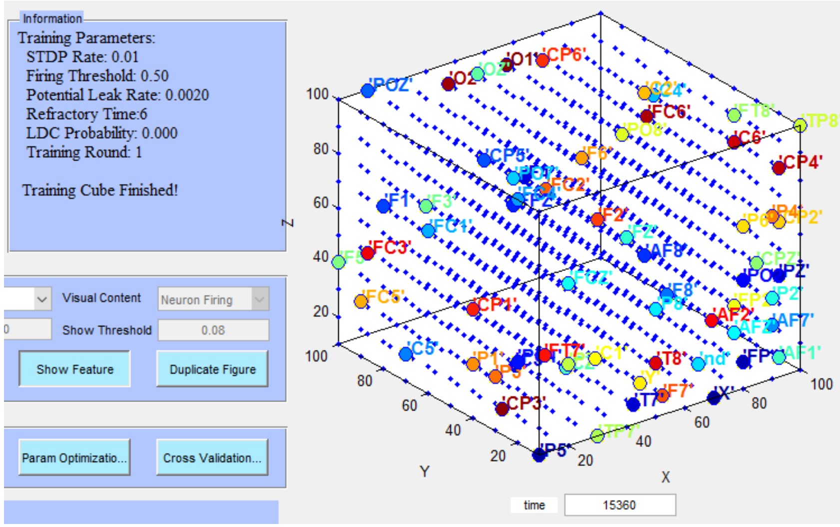
4. Training of the Cube

The next step is the unsupervised training of the Cube that will create connections between the neurons based on the input spikes.

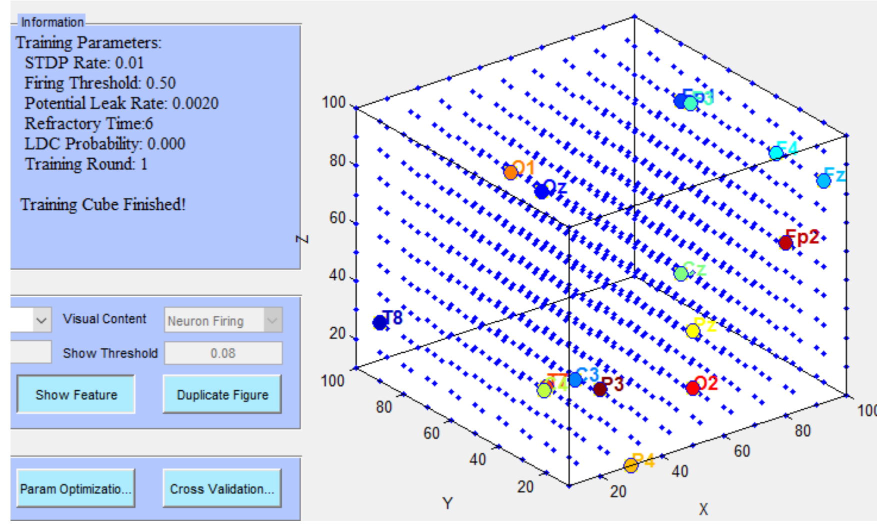
The parameters have the following meanings:

- **Potential leak rate**: the leak in membrane potential of a spiking neuron when the neuron does not fire
- **Threshold of firing**: the threshold membrane potential beyond which the neuron fires a spike
- **Refractory time**: the absolute time during which the neuron will not fire
- **STDP rate**: the learning rate of the STDP learning
- **Training round**: the number of iterations for unsupervised learning in the cube
- **LDC probability**: the probability of creating a long distance connection

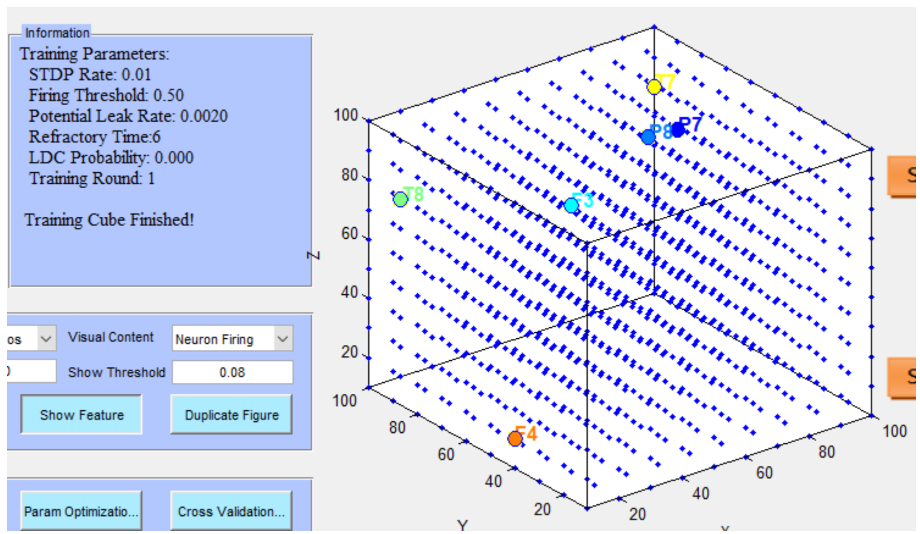




64 channels



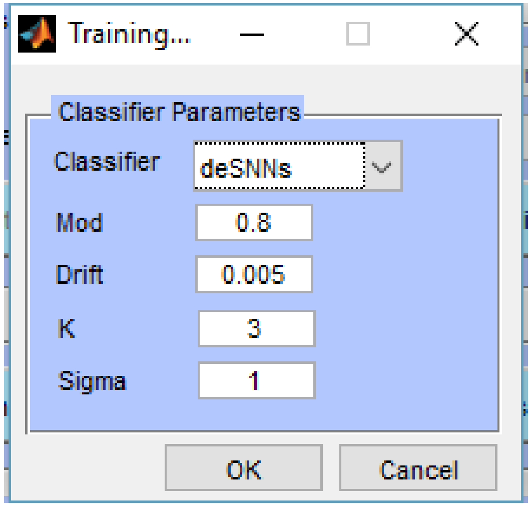
16 channels



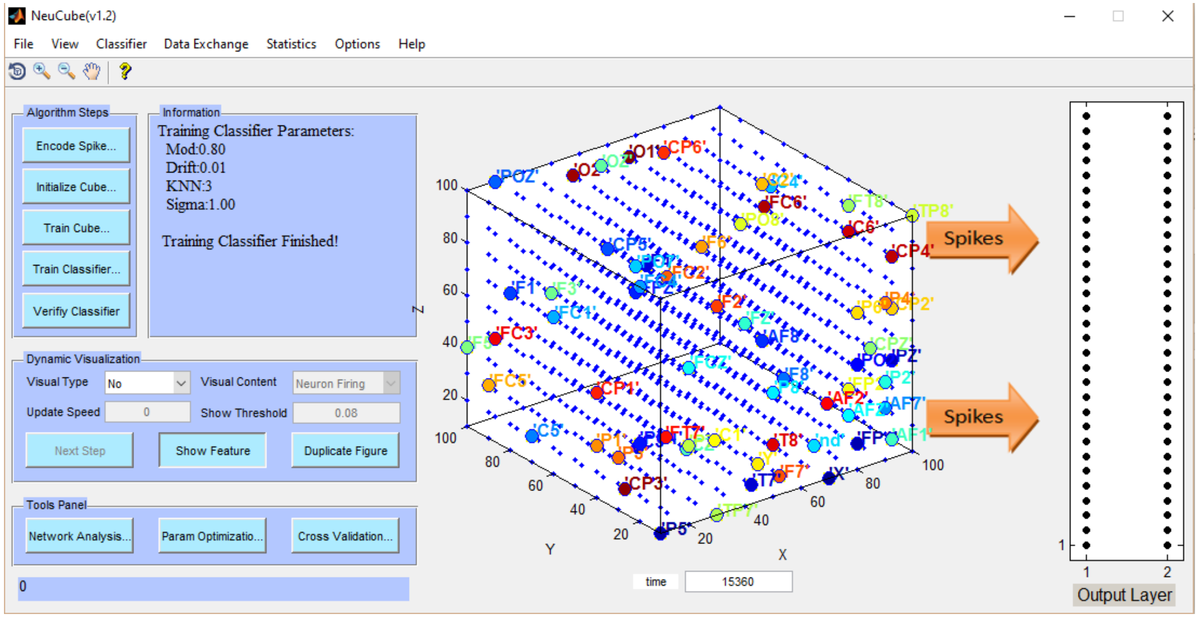
6 channels

5. Train Classifier

This step trains a model that takes the output spikes of the trained Cube as input and performs supervised learning to perform classification.

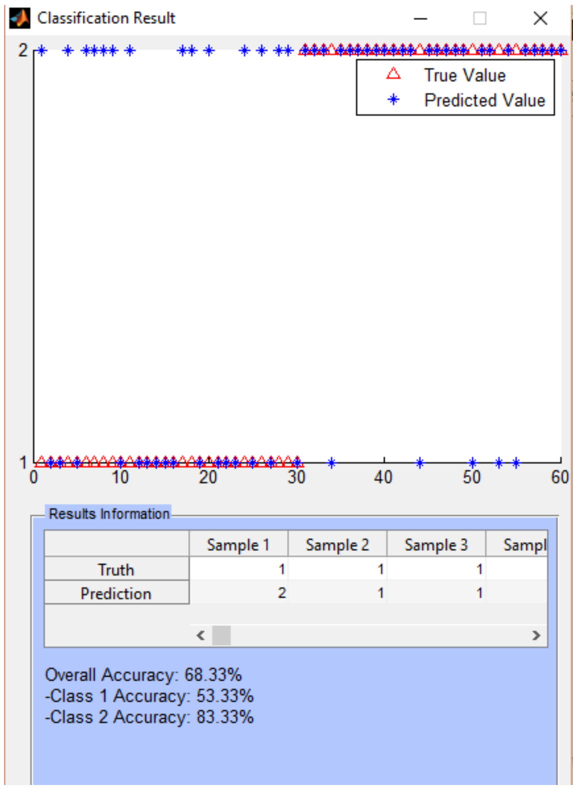


- **Mod** is a modulation factor, that defines how important the order of the first spike is
- **Drift** is a long term of the synaptic weights, depends on the value of the weight itself

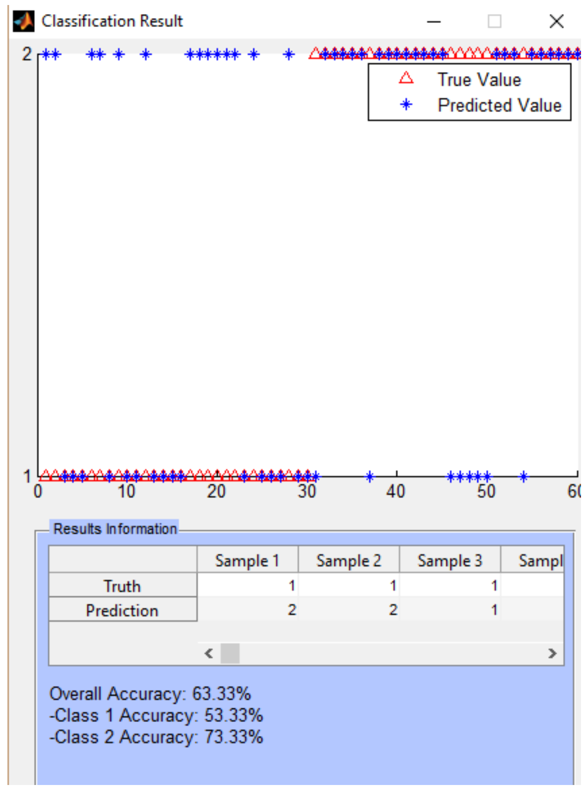


6.Verify Classifier

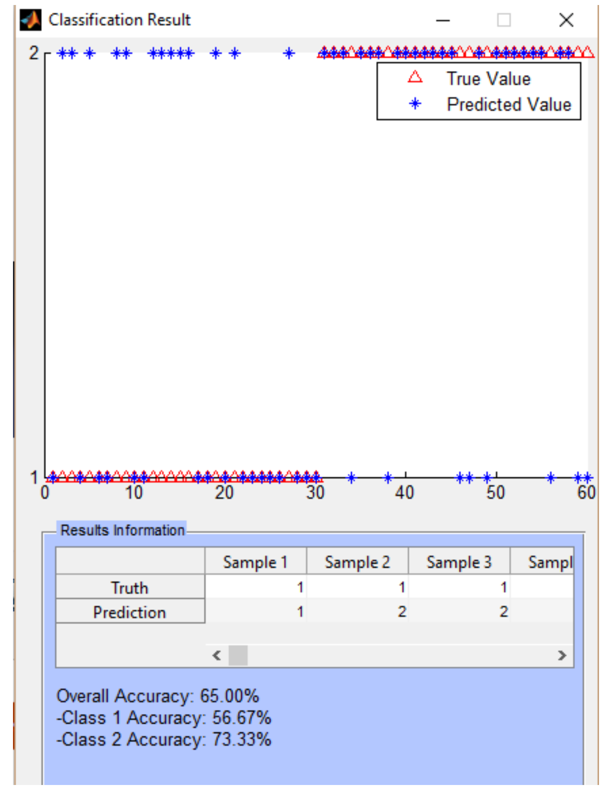
This step is used to verify the accuracy of the model built by deSNN learning. Clicking on “*Verify Classier*” begins the verification procedure. The 2D plot shows the sample ID against the class label, and the legend describes the actual and predicted class labels. The “*result information*” table lists these labels for each sample. Overall and class-wise accuracies are shown at the end.



64 channels



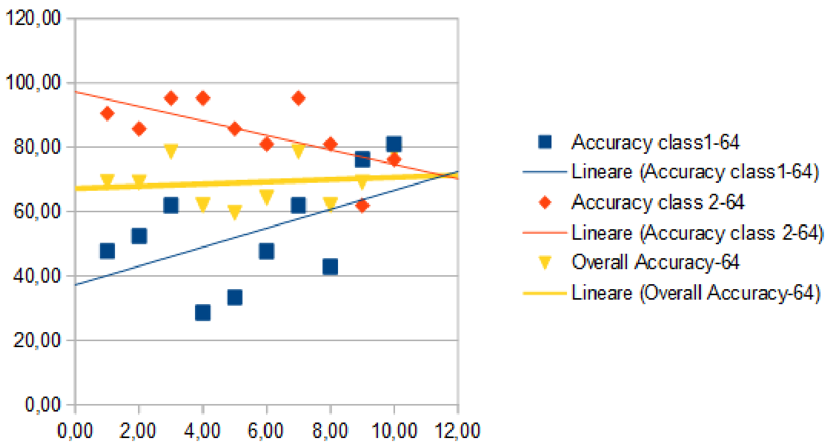
16 channels

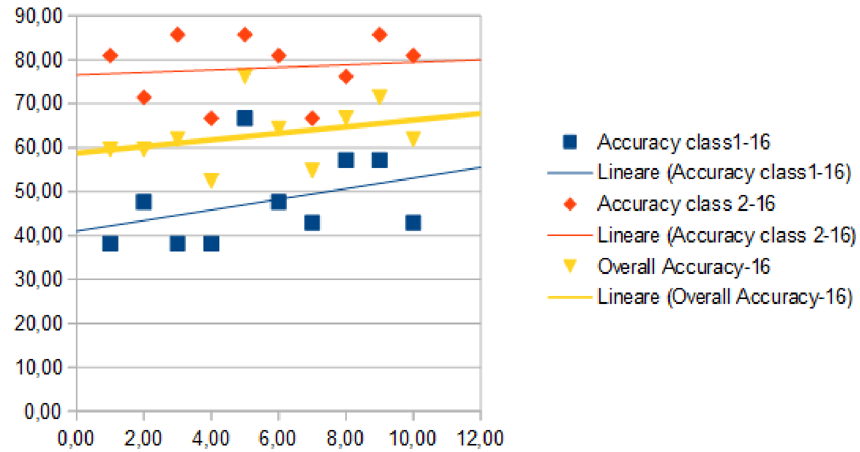


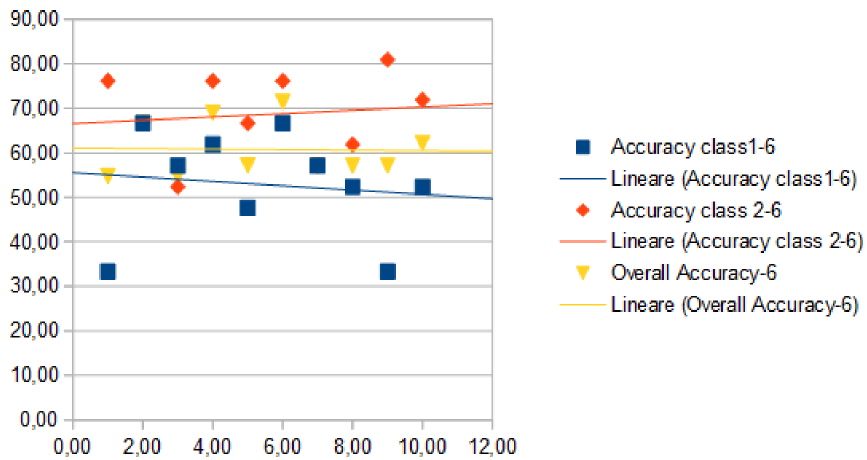
6 channels

7. Mean and Standard deviation

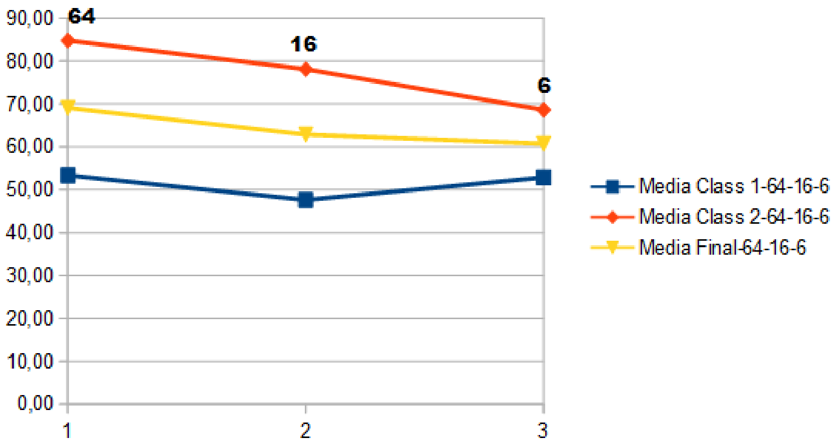
We tried to contain the problem of large error classifier on our dataset. To do this we conducted 10 tests for each view of the previous cases, so we could get a more reliable average of 64,16 and 6 channels.

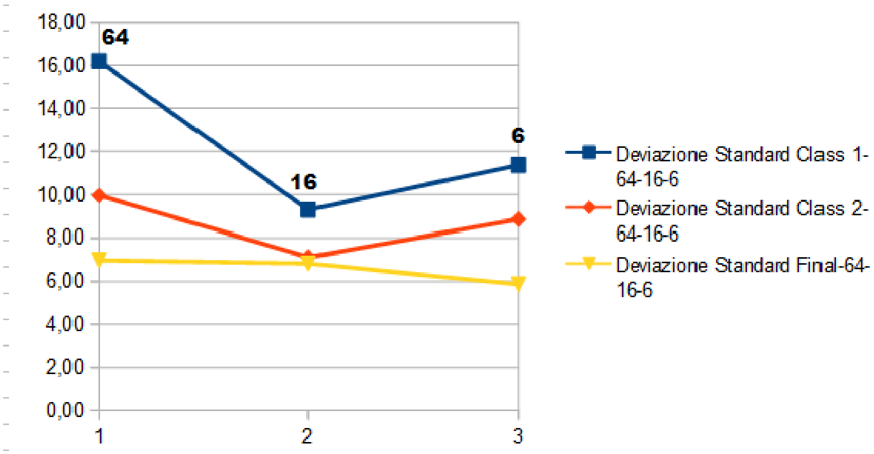






Once the trend lines are obtained we compared the results with each other in a graph for means and another for the standard deviations on class 1, class 2 and overall accuracy for 64,16 and 6 channels.

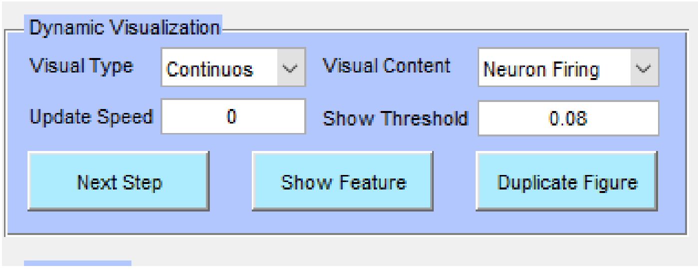


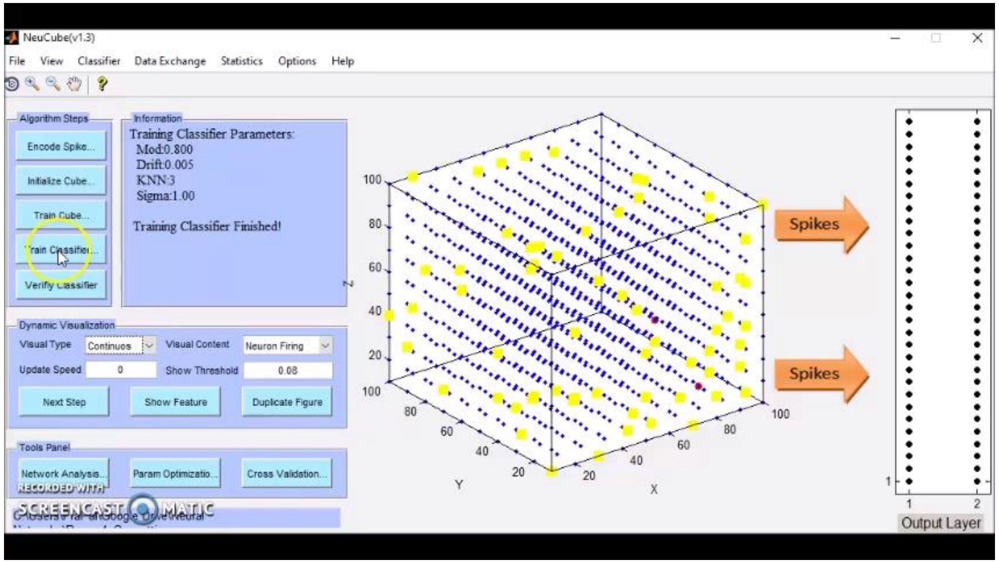


The results show that the best accuracy is given by tests on 64 channels, while the standard deviation obtains a slight improvement using only 6 of these.

Visualization Analysis

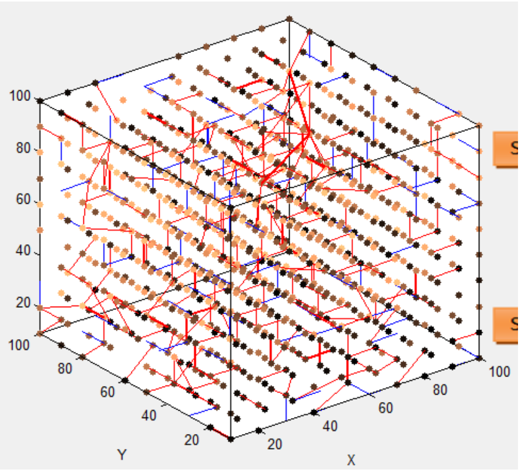
The unsupervised learning process can be visualized dynamically while the system is learning or can be saved as a movie for later usage and analysis by using the “dynamic visualization” panel

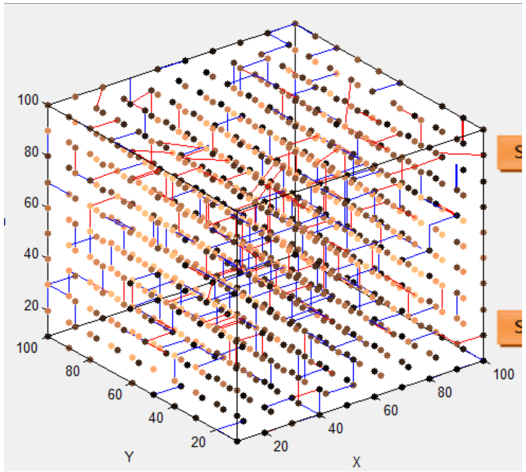


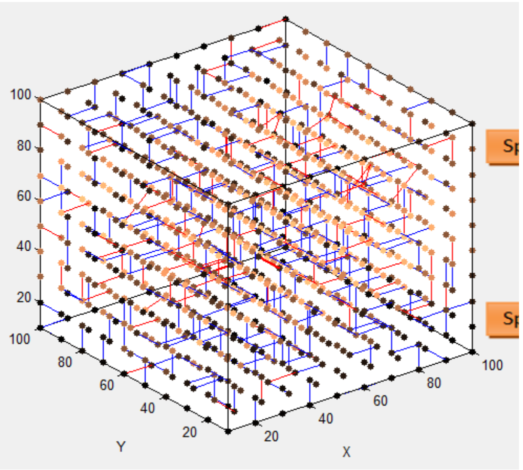


1. Analysis/visualization of the SNNcube connectivity

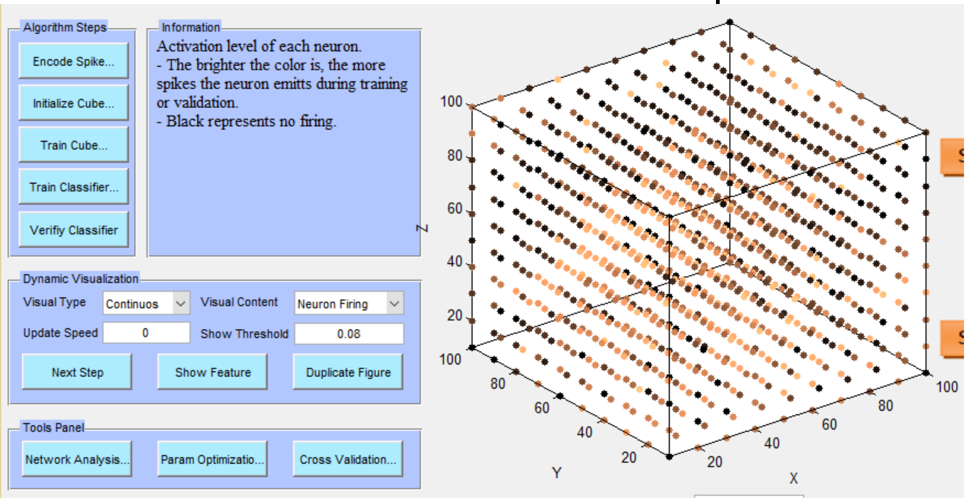
1. Clicking "*Show Connections"* and choosing a threshold displays the connections above a threshold value



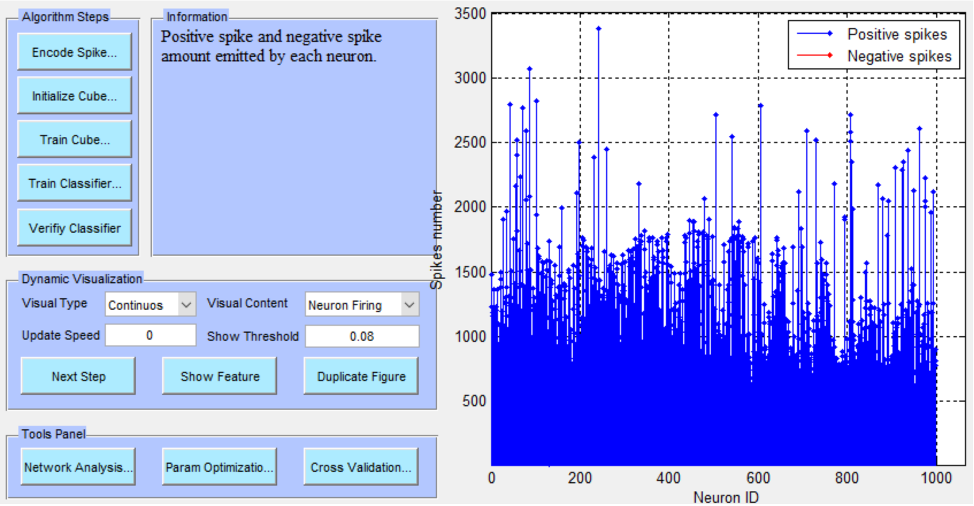




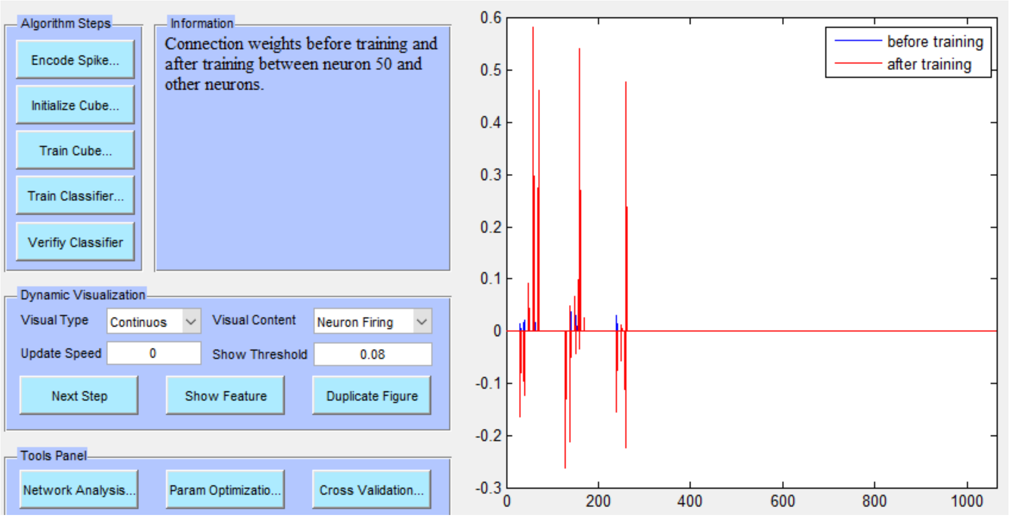
2. Clicking "*Activation Level*" shows the membrane potential of the neurons.



3. Clicking “*Spikes Emitted*” shows a histogram of positive and negative spikes emitted by all Cube’s neurons



4. The “*Neuron Weight*” option allows the user to specifically choose a neuron ID and visualize the connection weights of all neurons connected to the chosen neuron

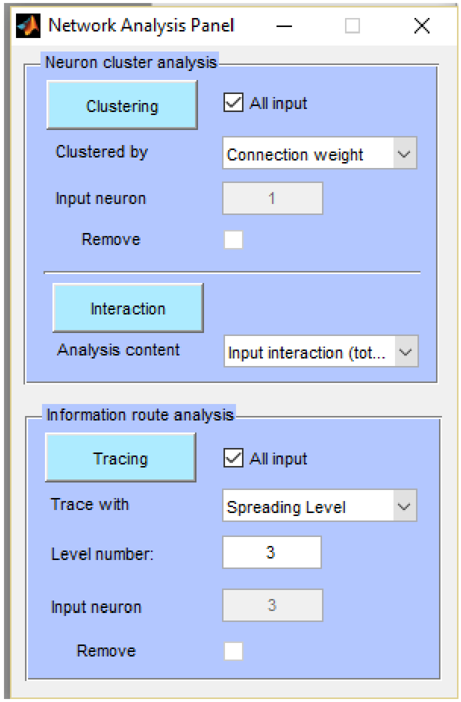


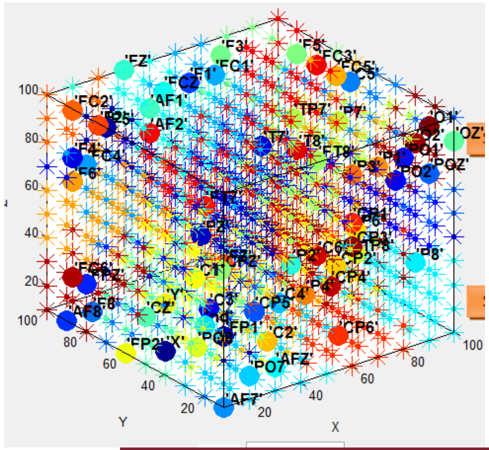
2. Analysis through the network analysis panel

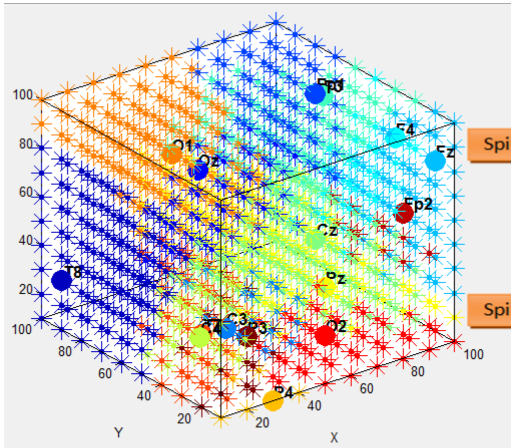
Analysis of the learned Cube network can be performed using the network analysis toolbox, which is initiated by clicking the “*Network Analysis*” button in the tools panel.

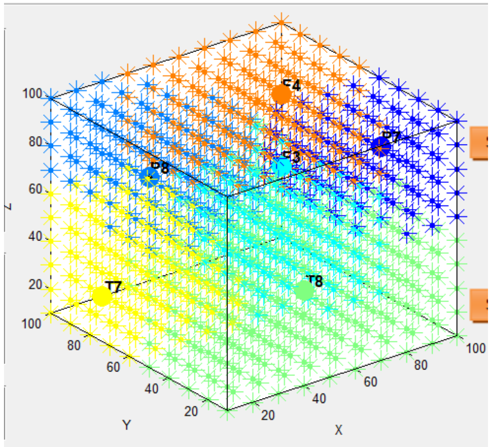
Different types of analysis are provided:

- **Neuron cluster analysis**: is used to analyze clusters of neuron-surrounding input neurons. The clustering is done by the connection weight which is the synaptic weight between a pair of neurons. It is adjusted during unsupervised learning to reflect the interaction between the neurons.

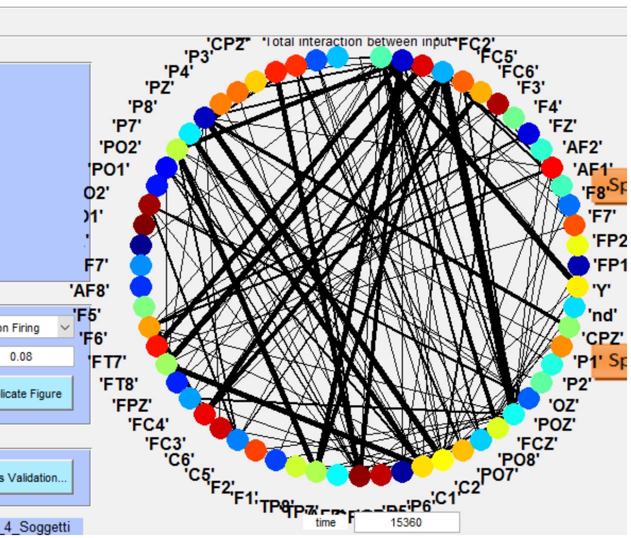




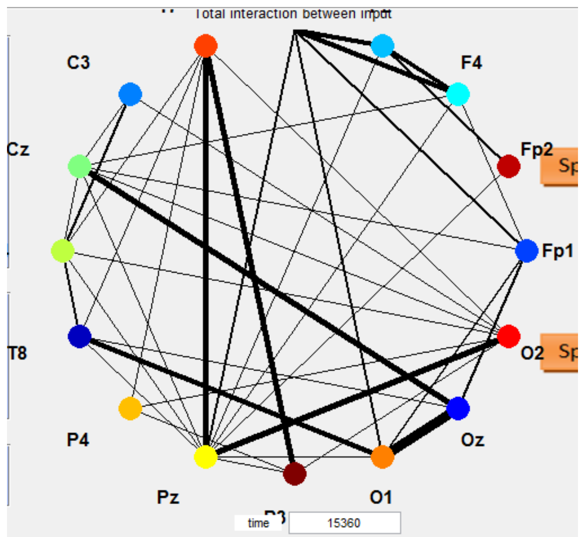




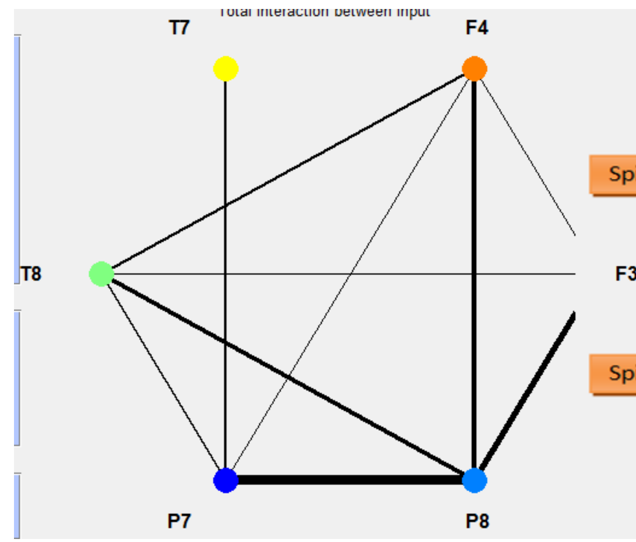
- **Interaction Analysis:** Interactions are analyzed using the following metrics:
  - *Input interaction (total)*: the total interaction between the input neuron clusters given by the cluster analysis
  - *Input interaction (average)*: the average interaction between the input neuron clusters given by the cluster analysis
  - *Neuron proportion*: the percentage of neurons in the cube which belong to an input neuron cluster



64 channels



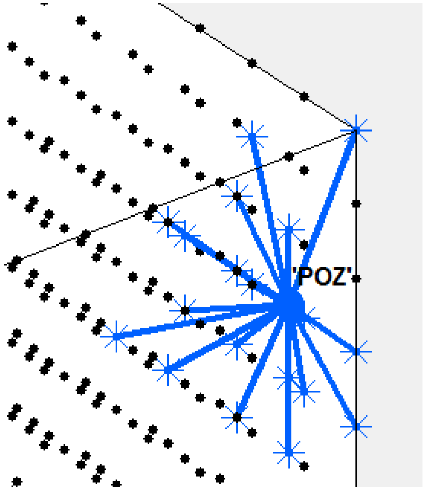
16 channels

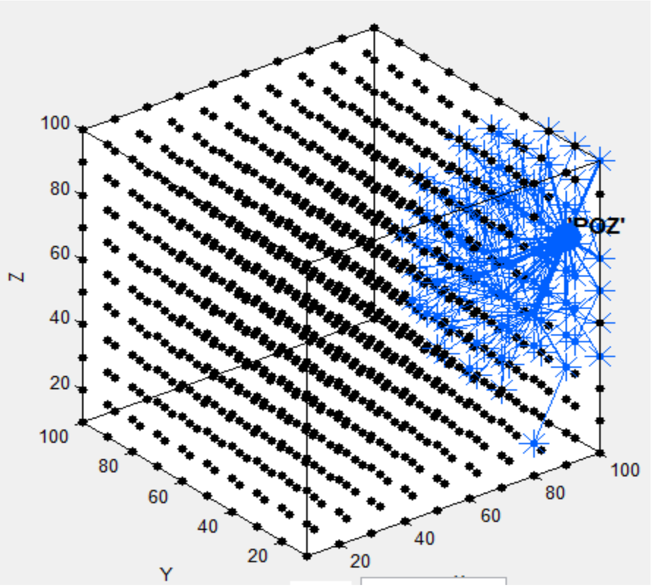


6 channels

- **Information route analysis**: is used for analyzing the information propagation route of the spikes. Different methods of analysis are available:

  - *Max spike gradient*: shows a tree rooted by the input neuron, a child neuron is chosen to be connected to a parent neuron if it receives spike from its parents

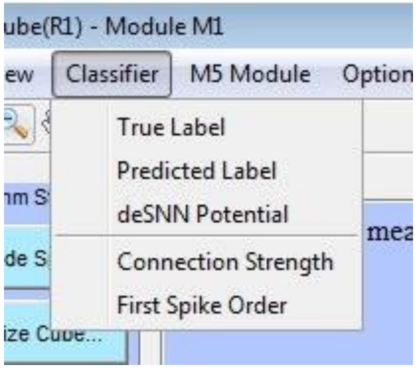




  - *Spreading level* : shows a tree from the input neuron to its neighborhood which reflects the spreading of the spikes.

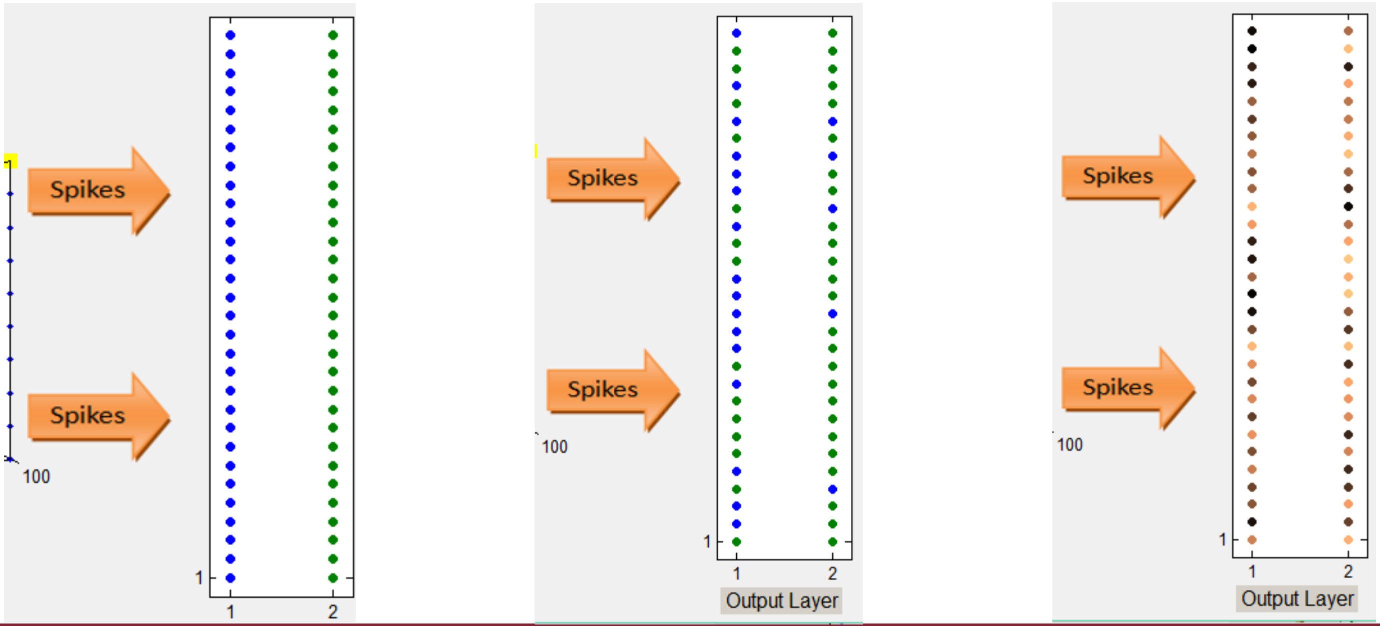
- *Information amount*: shows a tree rooted by the input neuron where a child neuron is chosen to be part of the tree only if it receives a minimum percentage of spikes from its parent neuron.

3. Output layer visualization

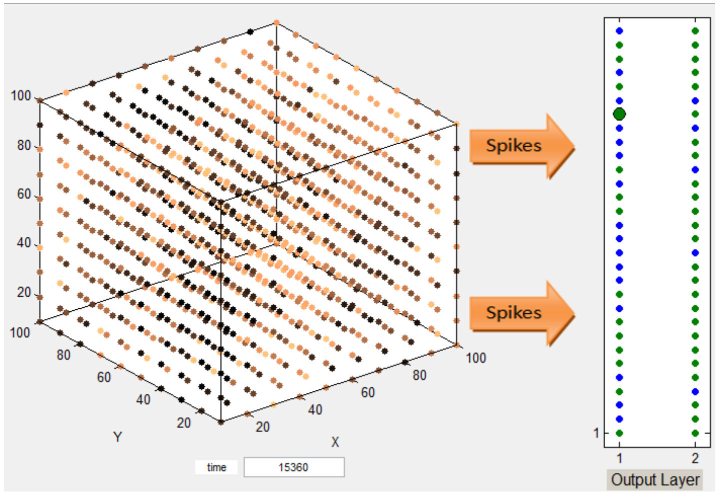


This option offers five methods for analysis.

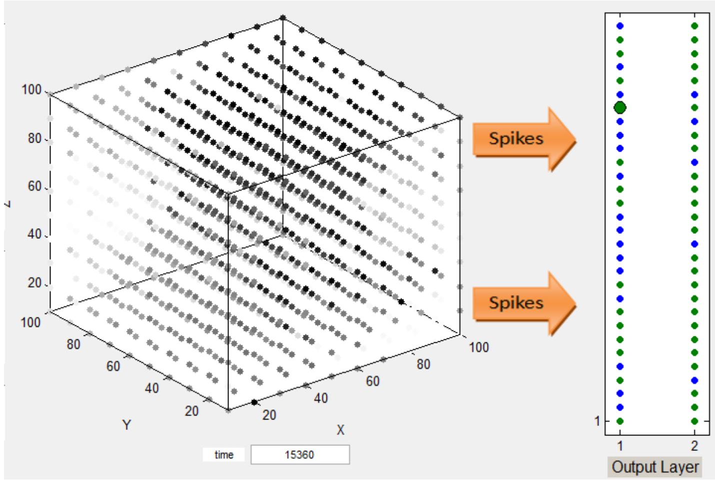
- *True label* : displays the true label of each sample by using a different color for each class. The samples are ordered by their number from bottom to top.
- *Predicted label* : displays the predicted label of each sample from the test/validation data set in the same way as for the true labels.
- *deSNN potential* : displays the membrane potential of the output neuron per sample. A brighter neuron signifies higher potential.



- *Connection strength*: enables the user to visualize the strength of connections between the neurons for every output neuron. Brighter neurons are more strongly connected than darker neurons



- *First spike order* : enables the user to visualize the spiking order of the neurons for each output neuron. Brighter neurons fire earlier than darker neurons.



Improvement of the Accuracy

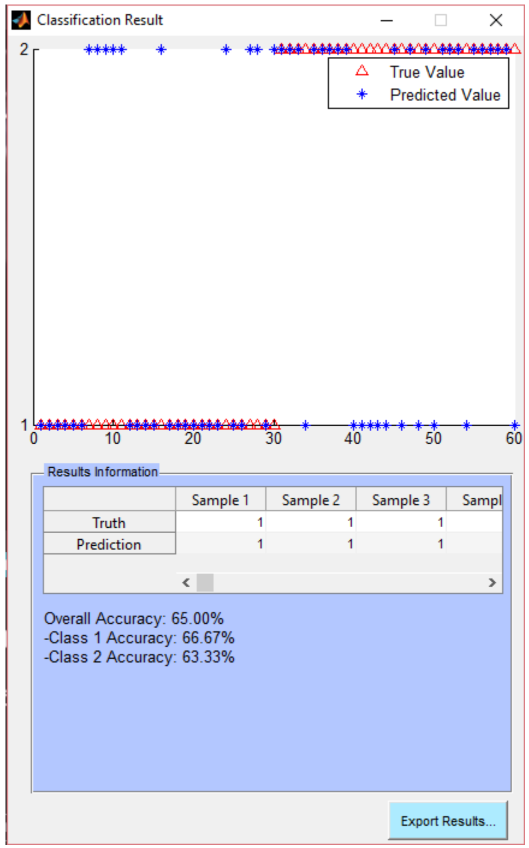
Some modications to the used data are taken upon, since the accuracies of the classier are not optimal:

1. Increase the number of training iteration
2. Optimize the parameters
3. Extend the number of people from 4 to 20

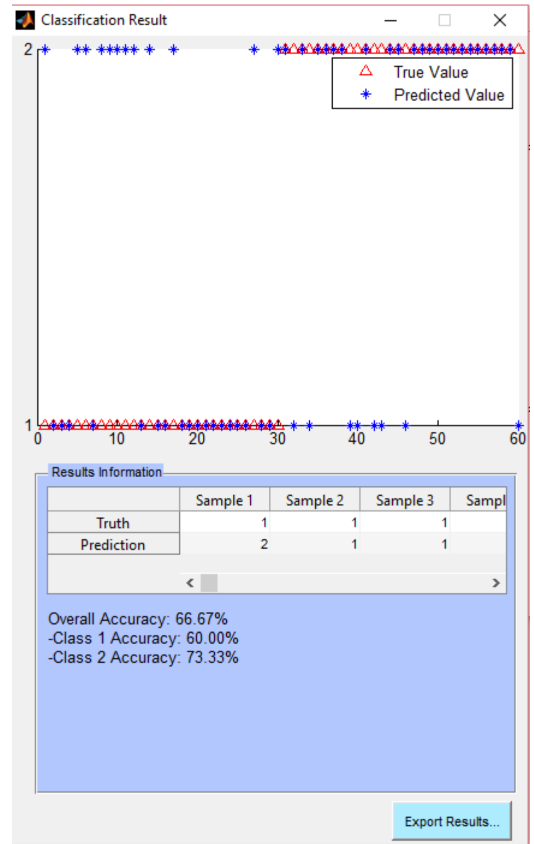
We worked on the dataset with 16 channels to have some kind of midway results.

1. Changing the number of Iteration

We tried to obtain better accuracy by improving the number of training rounds for the unsupervised learning of the cube. We redid the training for the dataset with 16 channels and obtained an accuracy of the classifier of 58,33%



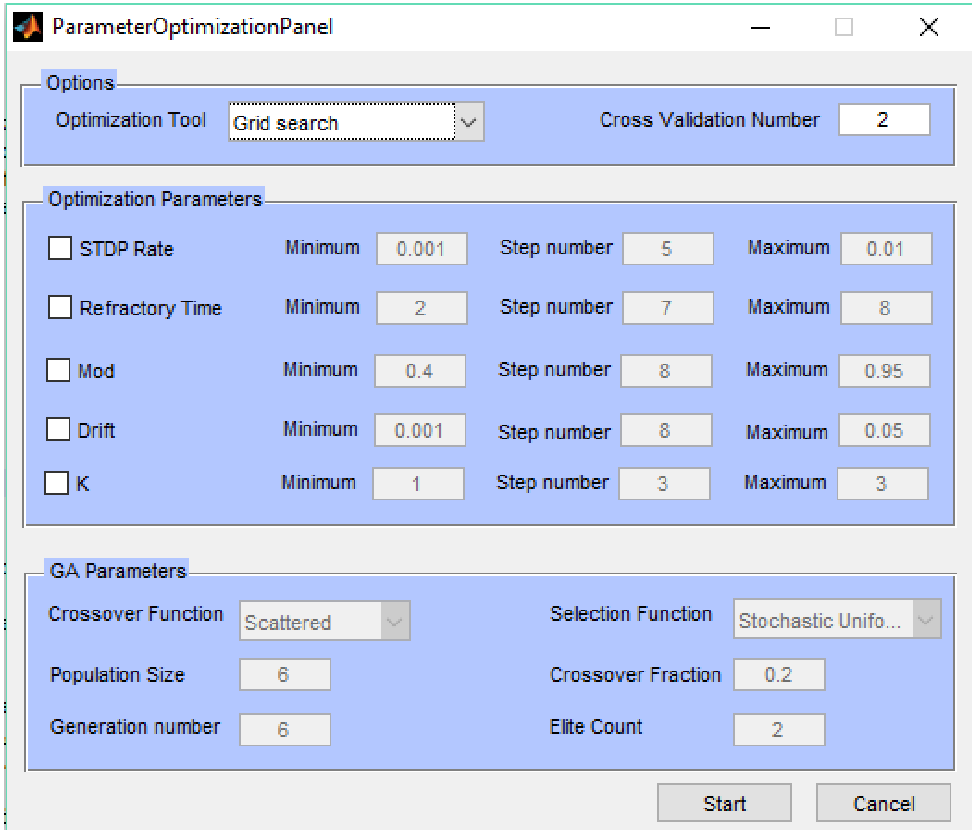
3 training iterations



5 training iterations

2. Optimization of parameters

Parameter optimization can be used to search for an optimal set of parameters that minimizes the test accuracy of the model. The computational time for parameter optimization depends on the number of parameters to be optimized and the size of the NeuCube model.



Exhaustive grid search is an exhaustive search method using a grid-based combination of parameters. The "*Optimization parameters*" subpanel can be used to specify the parameters to be optimized by enabling the checkboxes. During this process the running time and estimated time remaining for the optimization can be visualized in information panel.

We redid the training for the dataset with 16 channels and obtained an accuracy of the classifier of 61,67%

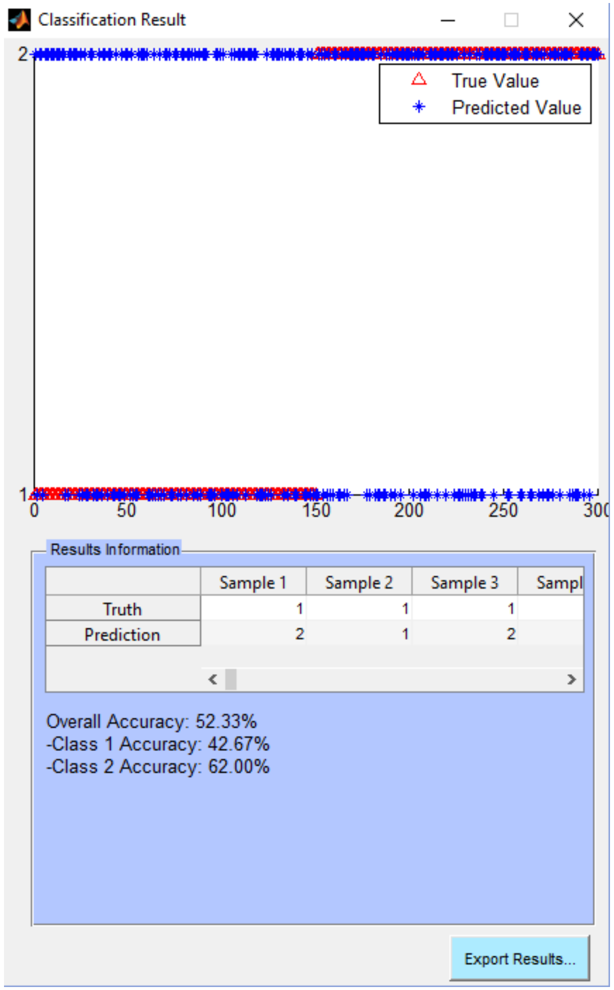
Since the parameters optimization is too slow to implement for all the 4 parameters altogether, we had to first use it for the STDP Rate and Refractory Time with step number equal to 10 for each one (160 minutes) and then we reused it for Mod and Drift (46 minutes).

```
Lowest Error: 0.36
AER  threshold:0.500, Small world radius:2.500, STDP rate:0.010,
Firing threshold:0.500, Refractory time:6.000, Train round:1.000,
deSNN mod:0.400,  deSNN drift:0.022
```

Using these parameters we obtained a overall accuracy of 68.33%, better accuracy but the process is too slow.

3. Extension of number of people

We tried to use of the total number of people of the dataset, which is 20 (10 control and 10 alcoholic). The implementation of the classifier is slower (350 minutes with parameter optimization) and the accuracy, even after the parameter optimization doesn't seem to be better than the previous one, rather it's worse.




```
Lowest Error: 0.43
AER  threshold:0.500, Small world radius:2.500, STDP rate:0.007,
Firing threshold:0.500, Refractory time:2.000, Train round:1.000,
deSNN mod:0.889, deSNN drift:0.001
```

Conclusions

In this project we have deeply analyzed the software NeuCube using a classification problem of an EEG Database. We have described step by step the instruction to correctly evaluate a classifier and to have some sort of visualization analysis of it.

This has been made by a confrontation of three different classifiers:

- the first one was formed by 64 channels
- the second by 16 channels
- the last one by 6 channels

What we have seen, is that the accuracy of all the three classifiers was not optimal and so we tried three distinct approaches to find a solution:

1. *Increase the number of training iteration* led to a better accuracy for both 3 and 5 training rounds
2. *Parameter optimization* was a slower process (around 200 minutes) but it was possible to obtain better results than other options
3. *Increase number of people* was the slowest one (350 minutes with parameter optimization) and didn't really improve the accuracy of the classifier.

The main problem with the software is the high error rate when the option “*Train Cube*” is selected, which give us low accuracy.

In the section "Mean and Standard Deviation" we seek a way to minimize this error but we have to state that the software used was still in beta version and only the module M1 was available.

In the future will be possible to achieve better results thanks to the implementation of the module M8 which will have all the functions of module M1 but it will also include specific functions that will support integrated brain data modelling, including EEG.