Big Data in simulation of brain activities

Big Data seminar

March 1, 2017

Preamble

- The workings of our brain is one of the most fascinating and challenging subjects for research and practical applications.
- During the last couple years, we've been working on a number of projects related to BCI – Brain Computer Interface.
- The results allow us to see the research opportunities in studying the cognitive processes in the brain, the ultimate Big Data machine.

Outline

The talk will touch on the following subjects:

Brain Computer Interface:

From electrical signals on the head's surface to recognition of neuron activities

Models of electrical field inside the brain's cortex:

Potential fields and method of fundamental solutions

Known inverse problem solutions in potential field models:

Mathematical challenges in finding a stable and satisfactory solution

Suggested approach:

Big data model and constructing Artificial Neural Network

Experiments:

Description of the process of gathering data

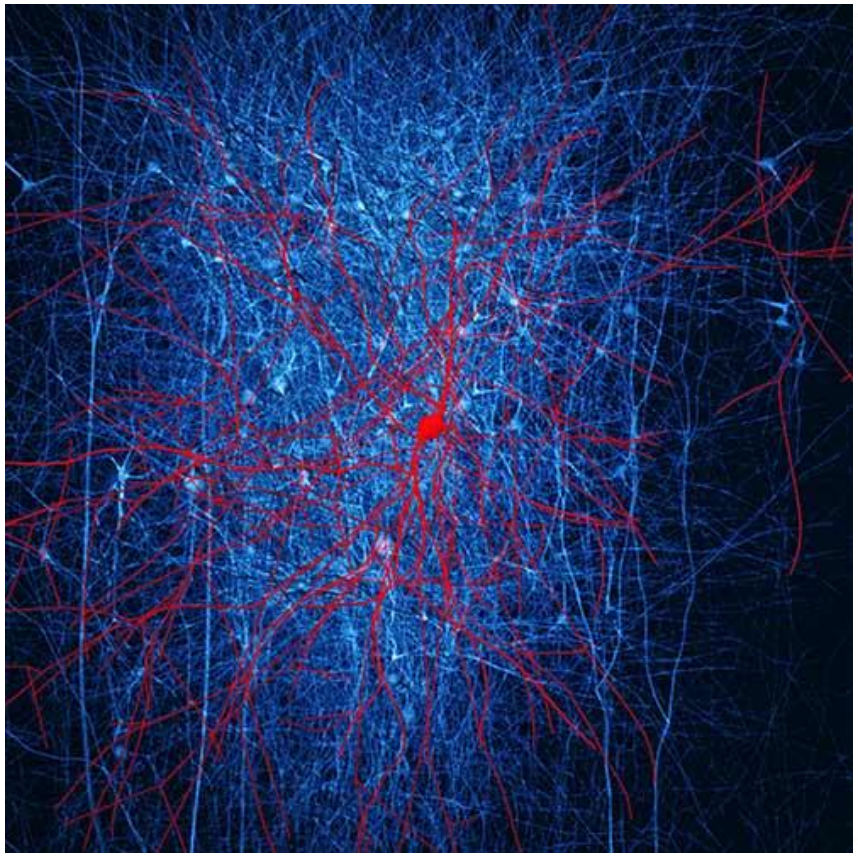
Machine learning applied to solution of the inverse problem

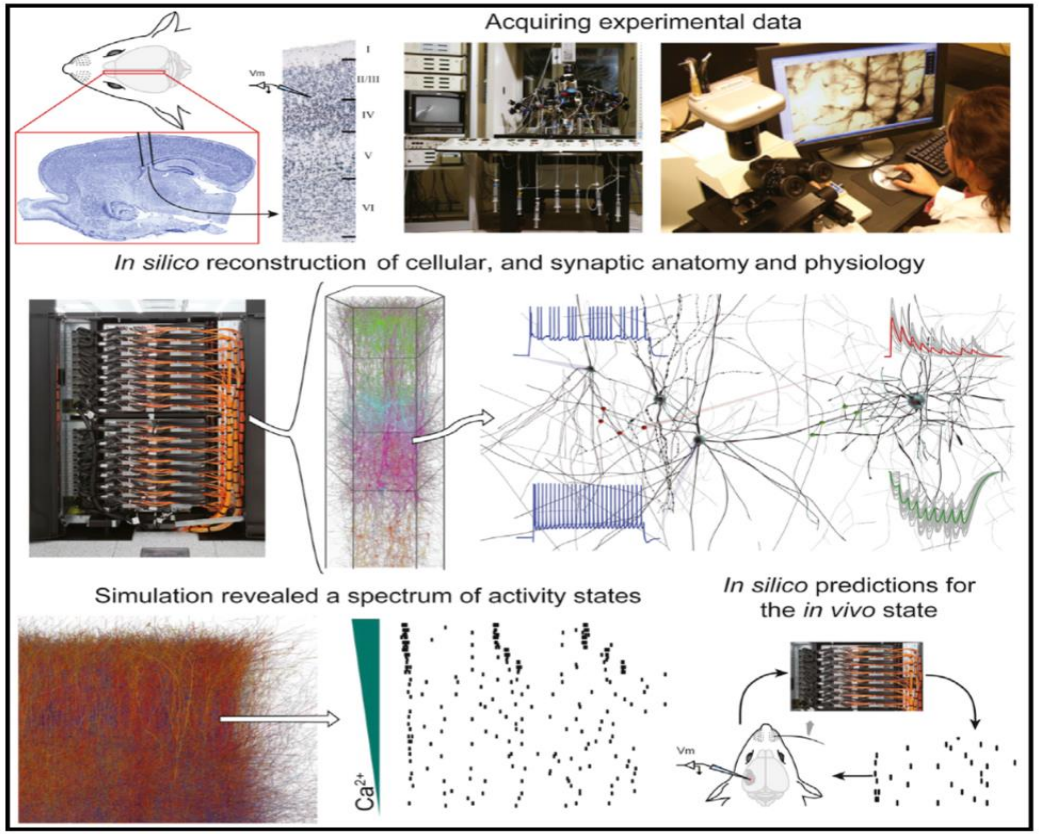
Supervised and unsupervised models and Deep Learning methods

What do we know about brain…

Human Brain *in silico*: the Big Data model

"Reconstruction and Simulation of Neocortical Microcircuitry", published in *Cell* on Oct. 8, 2015



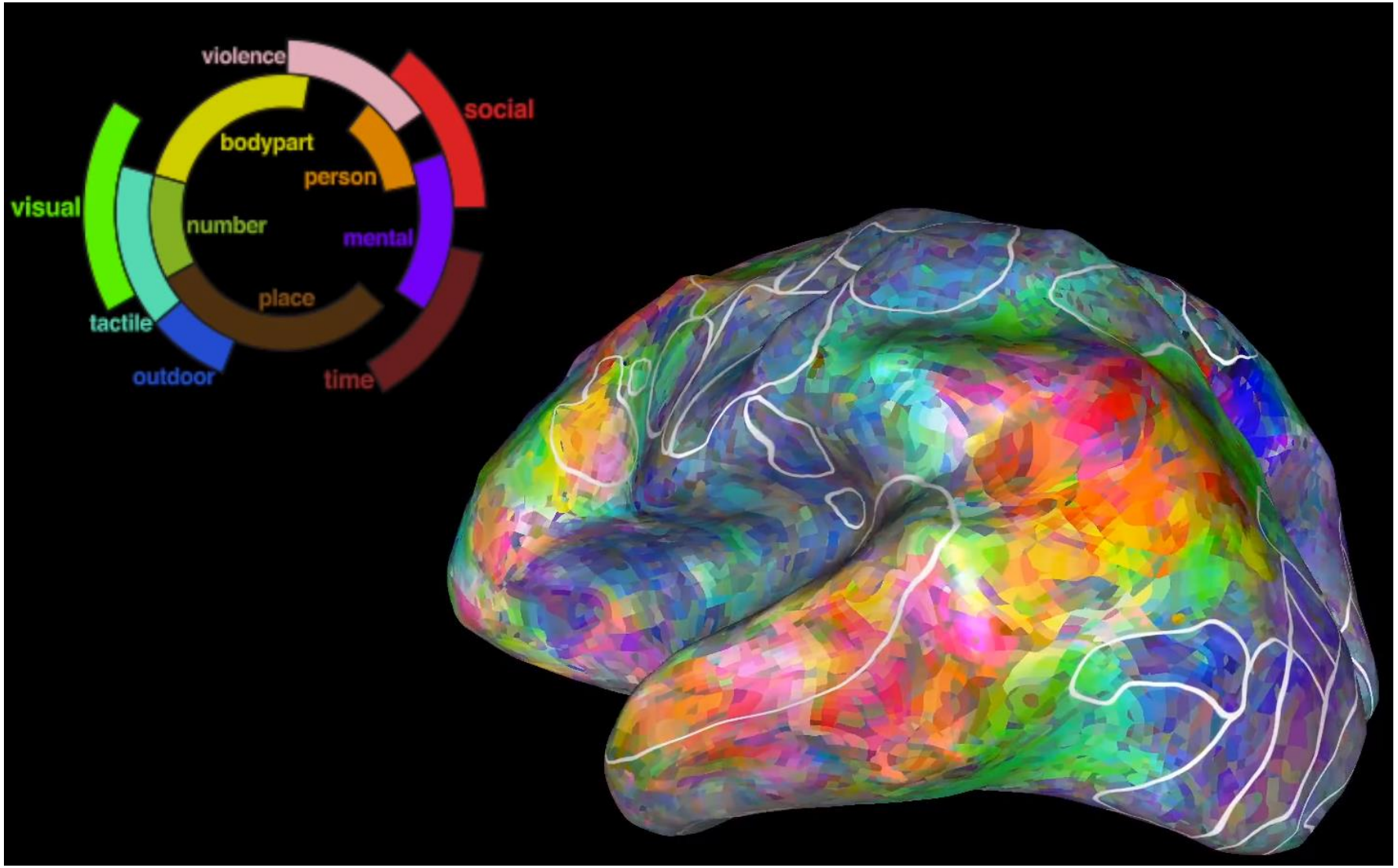


It's a piece of rat brain containing about 30,000 neurons and 40 million synaptic connections.

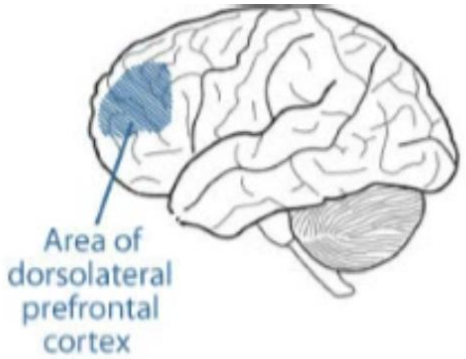
There's nothing remarkable about it, except that it isn't real!

It's a digital reconstruction—a representation of a one-third cubic millimeter of rat neocortex—and it seems to work like the real thing.

Brain’s toponymics



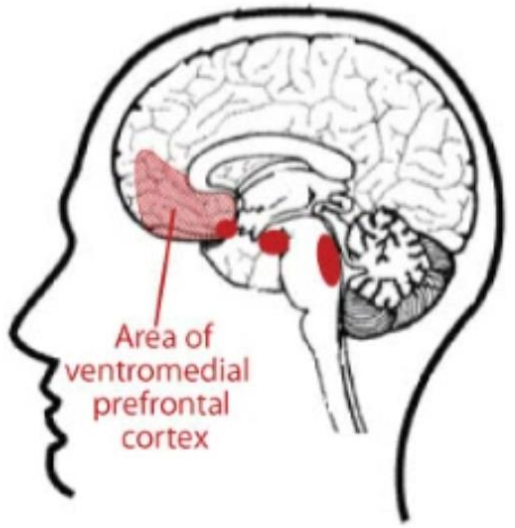
Cerebral cortex



The newer portion of the cerebral **cortex** serves as the center of higher mental functions for humans.

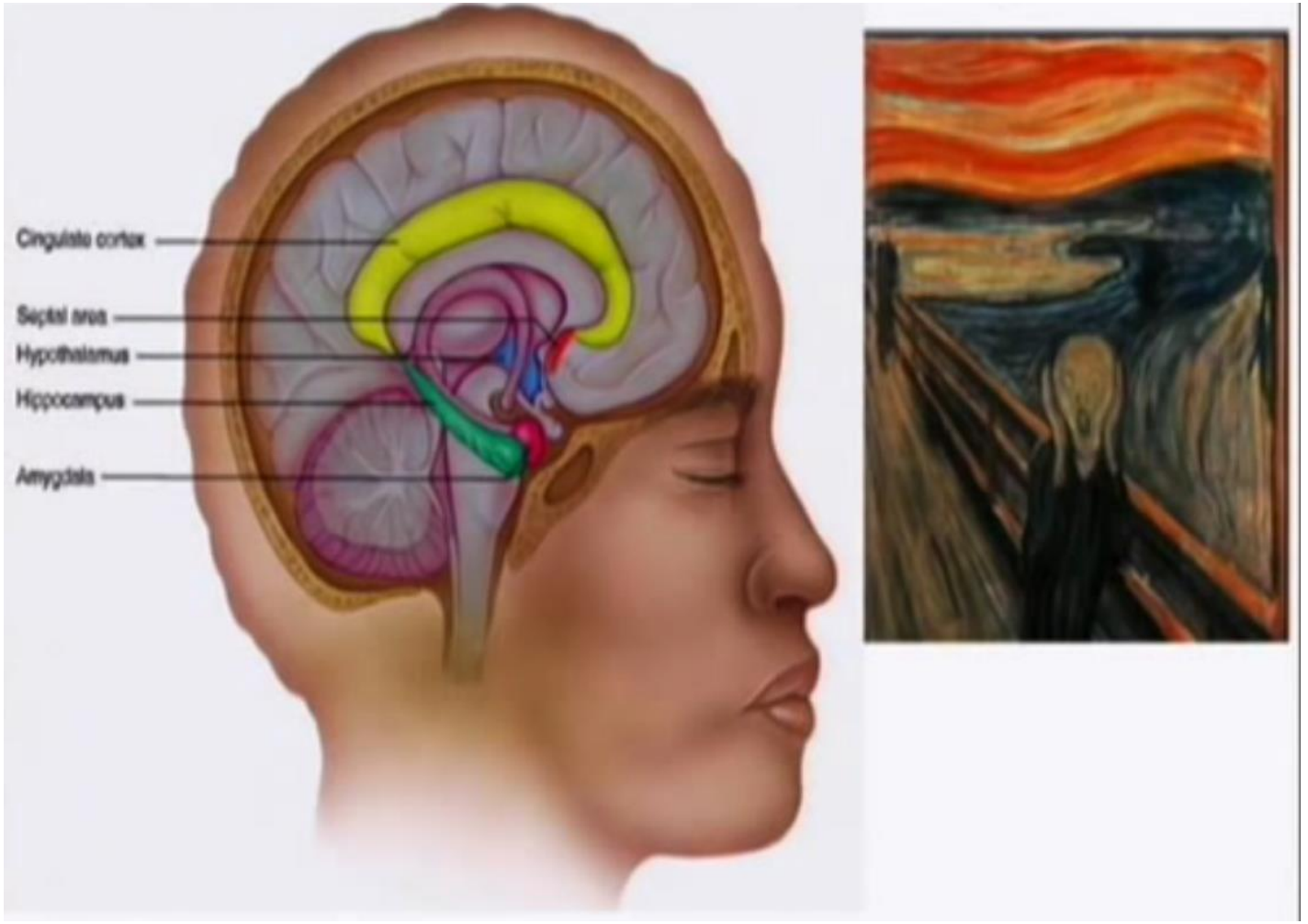
Some Functions of the Ventromedial Prefrontal Cortex

- Monitors brain and body information indicating emotional state
- Helps to assess emotional relevance and emotional viability of prefrontal planning
- Relays results of prefrontal processing to the amygdala, other emotion centers, and other brain areas.

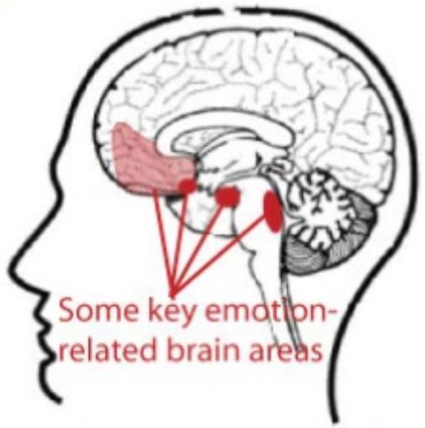


The neocortex contains some 100 billion cells.

Emotional brain



The spectrum of emotions



BACKGROUND FEELINGS

Irritation Discontent Grumpiness

Big Data in the eyes of the brain researchers

- The Open Connectome project
- The Activitome study
- The BRAIN Initiative

"… a mouse brain contains $500x10^9$ cubic micron pixels and if we want to record all of them for 20 min at 1000Hz we have 500 petabytes of raw data…" but

"… the big data could be … compressed into information albeit complex." *Prof. F. Engert, Harvard Univ.*

The information compression comes with simulation of the brain activities rather than of the brain topology.

It makes the Big Data model more manageable and productive.

How to collect Big Data for the Brain activities?

In-vivo – Live experimental methods

- Stimulate different emotional states,
- Collect the EEG information, or run fMRI
- Interpret the data collected, using
  - Signal processing
  - Frequency analysis,
  - Fractal analysis, and others

In order to map activated parts of the cortex onto the field of emotion

Inverse model solution is needed:

from EEG to the brain activated areas

Neuron dynamics

The detailed study of singular neurons is an interesting subject per se but most important is to understand

- *the macroscopic dynamics and*
- *function of neuronal networks*.

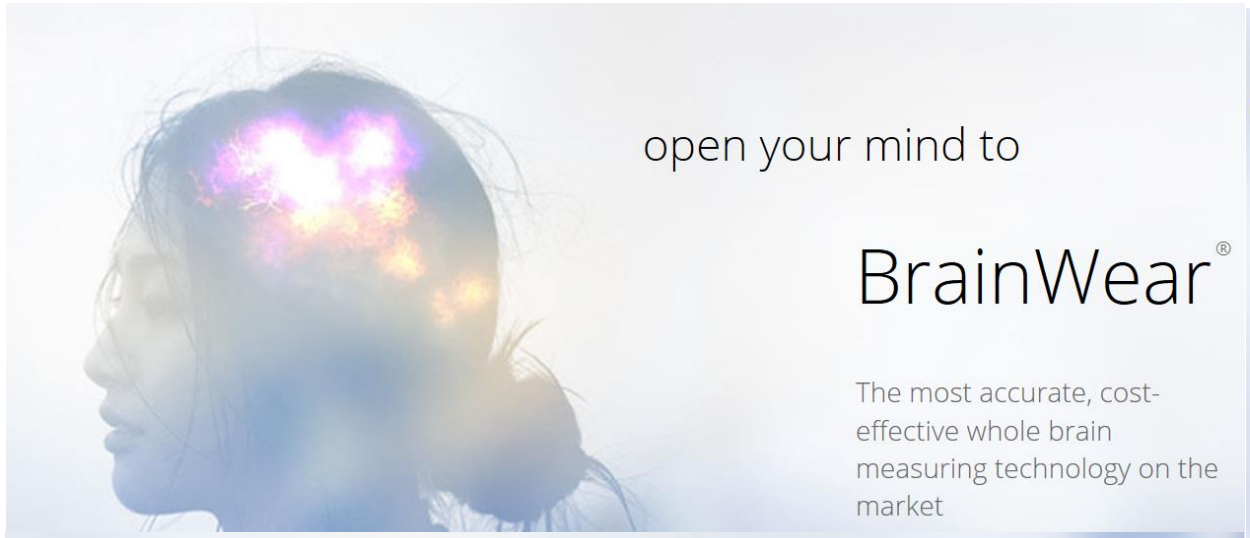
Neuronal networks present collective phenomena, like

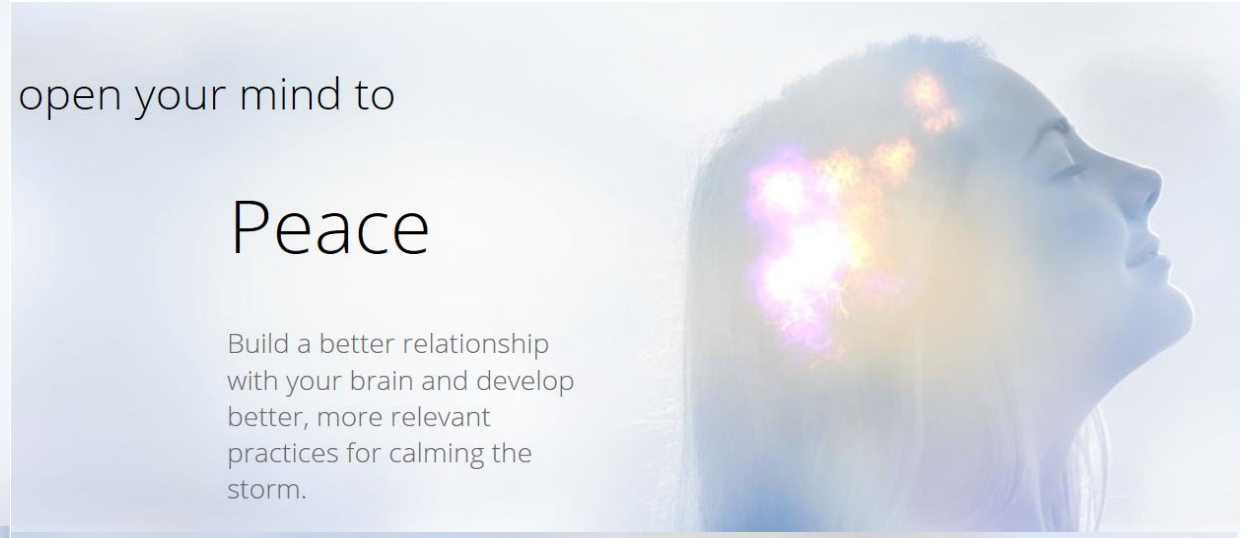
- *synchronization*, *waves and avalanches*,
- *global bifurcations* or *phase transitions*,

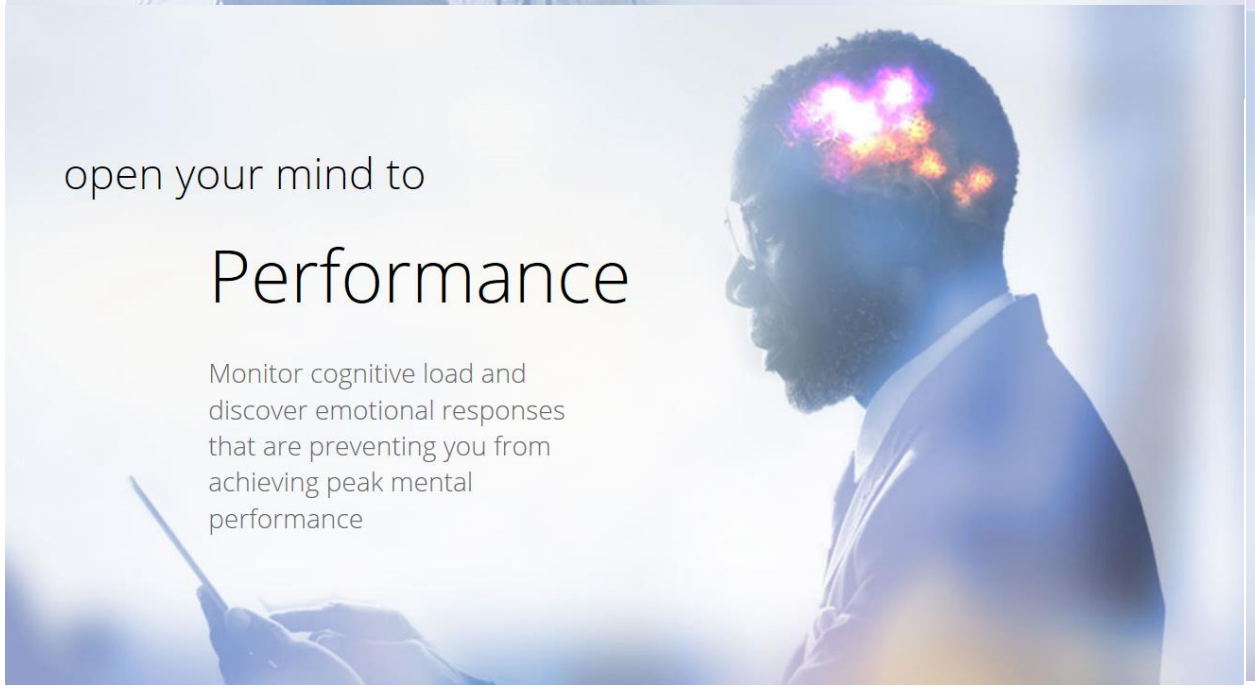
All of those cannot be studied in small neuronal populations.

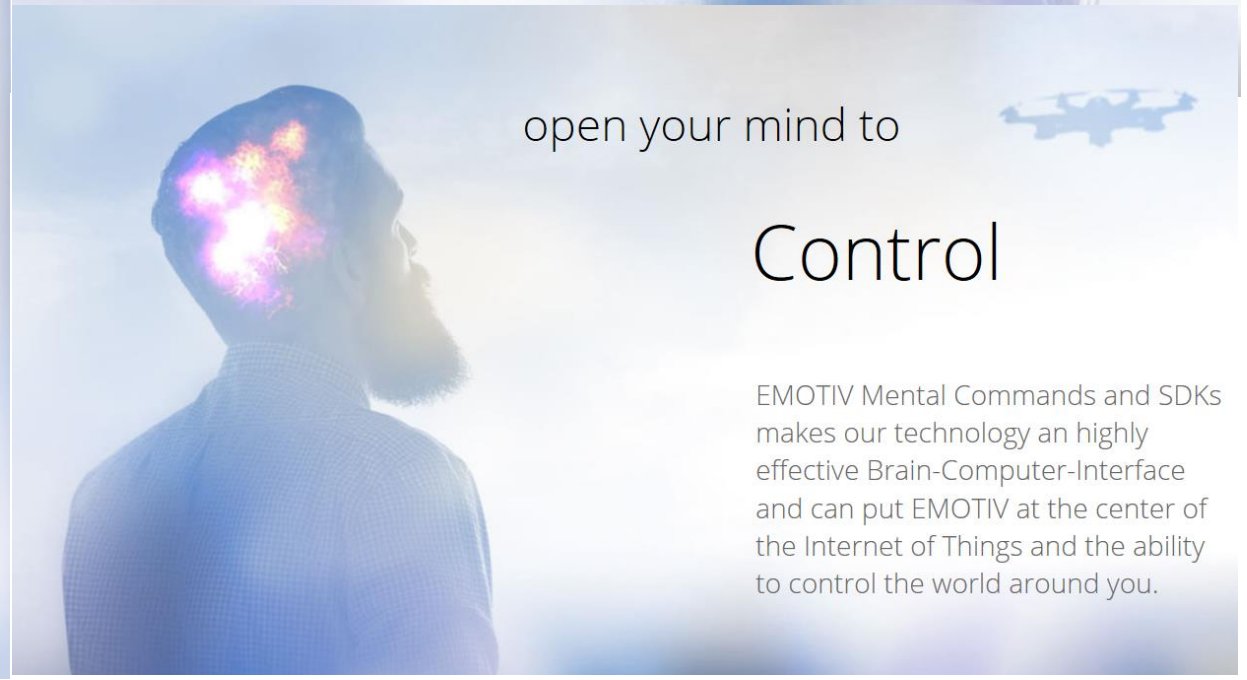
Brain Computer Interface

Experiments with eMotiv – a brainwear to register EEG









Experiments with eMotiv device

- The work started in Spring 2015 as part of the HCI class
- A few team projects by the graduate students were performed
- The eMotiv device was used to study BCI

Project 1 Emotional response to the pictures

Implications

At its basic level, a circuit takes an electrical input, deciphers a pattern and then – based upon the pattern and programming – the circuit opens and closed gates. The resulting configuration of gates allows the electrical signal to propagate in a certain manner thus, allowing for the signal to be processed and, in turn, that signal can produce an output – to be used as input – to a mechanical device.

This is pertinent to the topic in that; the EMOTIV device transmits an electrical signal – from your brain. So, simply thinking about something – that signal could be processes just as any another electric signal and be used to manipulate a circuit and/or computer chips. This means, you could think about moving a back-hoe and, if circuitry is configured correctly – the back-hoe will operate – based upon your thoughts!

The implications are astounding!

"Stephen William Hawking CH CBE FRS FRSA (born 8 January 1942) is a British theoretical physicist, cosmologist, author and Director of Research at the Centre for Theoretical Cosmology within the University of Cambridge." "Hawking suffers from a rare early-onset, slow-progressing form of amyotrophic lateral sclerosis (ALS), also known as motor neuron disease or Lou Gehrig's disease, that has gradually paralyzed him over the decades. He now communicates using a single cheek muscle attached to a speech-generating device."[2]

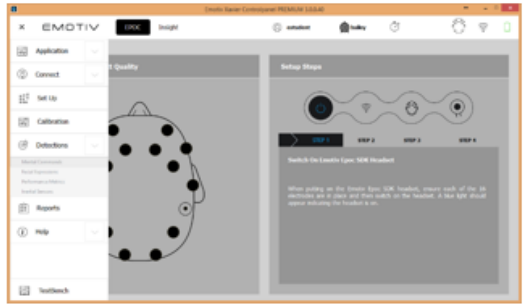
However, his mind is as active as ever and, with this sort of devise, Hawkins (and other paraplegics could regain motor functions – by, for example, wearing exoskeletons that they can control – with their mind!

Everything could become easier! This paper – it could write itself simply by me thinking about the text I want to enter – versus actually typing in the text.

The capabilities are only limited by our imagination!

SDK

The SDK is a software suite for the EMOTIV Xavier device.



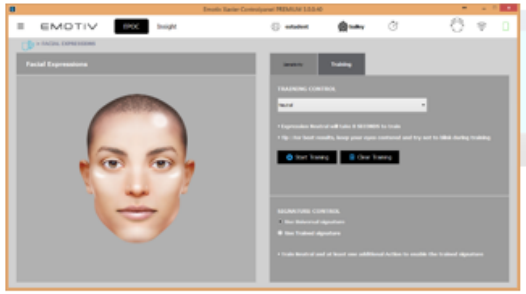
Calibration.

The first process it to calibrate the device/software to your specific signal. This is done though a two-step process: 1.) a training session where you wear the device and keeps your eyes open for about 30 seconds then 2.) you do the next 30 second trading session with your eyes closed. That's it!

EMOTIV XAVIER SDK

Detections.

The next set of training is a bit lengthier. This involves a series of mental comments (mentioned above) where you think about moving a 3D cube in 3D space.



Facial Expressions.

The SDK has the ability to detect facial expressions and, thus, could infer different non-verbal commands.

Performance Metrics and Internal Sessions are also included but will not be covered here.

Device/Composer.

What is the EmoComposser?

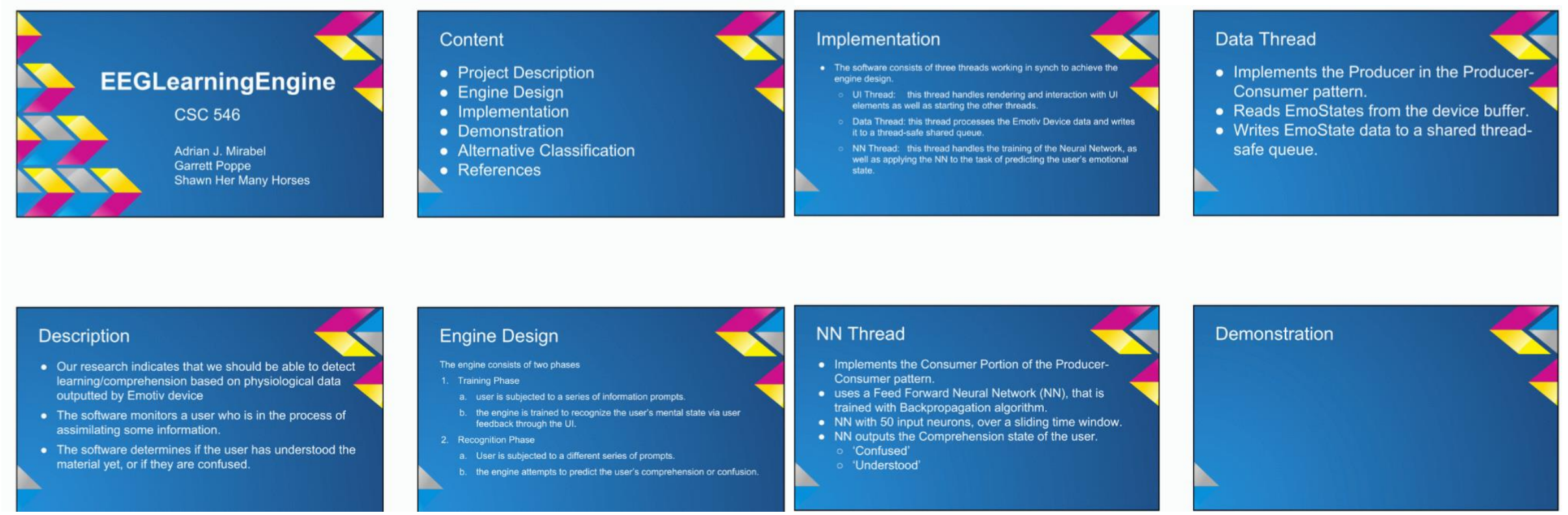
The EmoComposser allows the developer to program additional functionality, using the SDK, where – the composer supplies data to the SDK as though it were data from the device. This allows the developer to have a stable development environment – allowing for the signal to be repeatedly produced in a predictable manner.

Interactive component – pictured at left, creates a signal on each of the areas listed in the Detection tab. The label correlate with SDK methods – so, if you select: 1.) Auto repeat checkbox, 2.) Affectiv Excitement – to a positive number and then click 3.) "send" – the positive number selected (in the excitement area) will repeatedly be produced/send to the SDK on an interval of 0.25 seconds. The correlated code will be shown below, in the C# section.

EmoScripts component.

Once your program is running and accepting composer data, you can incorporate – EmoScripts. Scripts are similar to the Detection functionalist however, the methods produce pre-scripted/changing signals to the different methods. This would be used for further development –

Project 2 EEG Learning Engine



Project 3 Recognition of emotions in education – Reactions to different slides during a lecture

Table 1. Boredom

Participant	First Test	Second Test	Third Test	Average Time
Shadi Shiri	0.743	0.743	0.748	0.744
Anusha Karur	0.724	0.747	0.872	0.781
Manar Algarni	0.743	0.743	0.743	0.743
Mean	0.736	0.744	0.787	0.756
Standard Deviation	0.008957	0.001886	0.059668	0.017682

Table 2. Frustration

Participant	First Test	Second Test	Third Test	Average Time
Shadi Shiri	0	1	1	0.666
Anusha Karur	0	1	1	0.666
Manar Algarni	0.970	1	1	0.99
Mean	0.323	1	1	0.774
Standard Deviation	0.457	0	0	0.1527

Table 3. Excitement

Participant	First Test	Second Test	Third Test	Average Time
Shadi Shiri	0.0461	.407	0.615	0.371
Anusha Karur	0.423	0.547	0.081	0.350
Manar Algarni	0.477	0.608	0.237	0.440
Mean	0.315	0.520	0.311	0.387
Standard Deviation	0.1916	0.0841	0.22412	0.0384

The average time the participants were Bored during the experiment is 0.756 second. The average time the participants were Frustrated during the experiment is 0.774 second. The average time the participants were Excited during the experiment is 0.387 second.

Words, Emotions, Semantics – how they are imprinted in our brain

Identifying Emotions on the Basis of Neural Activation (Carnegie-Mellon U.)

"… Actors were asked to self-induce nine emotional states (anger, disgust, envy, fear, happiness, lust, pride, sadness, and shame) while in an fMRI scanner.

…. Using a Gaussian Naive Bayes pooled variance classifier, we demonstrate the ability to identify specific emotions experienced by an individual.

These results suggest a structure for neural representations of emotion and inform theories of emotional processing."

Tom Mitchel: How are words processed in the brain?

"We don't know. But … perhaps:

- Visual system recognizes character string and organizes to process word by 130 ms
- There is no single moment when the word is suddenly perceived
- Semantics trickles in over time
- Different cortical regions collaborate to jointly infer specific stimulus properties they specialize in
- Similar to other types of memory retrieval – an iterative, joint effort
- Left supramarginal gyrus serve as semantic hub collecting relevant semantics at 400 ms"

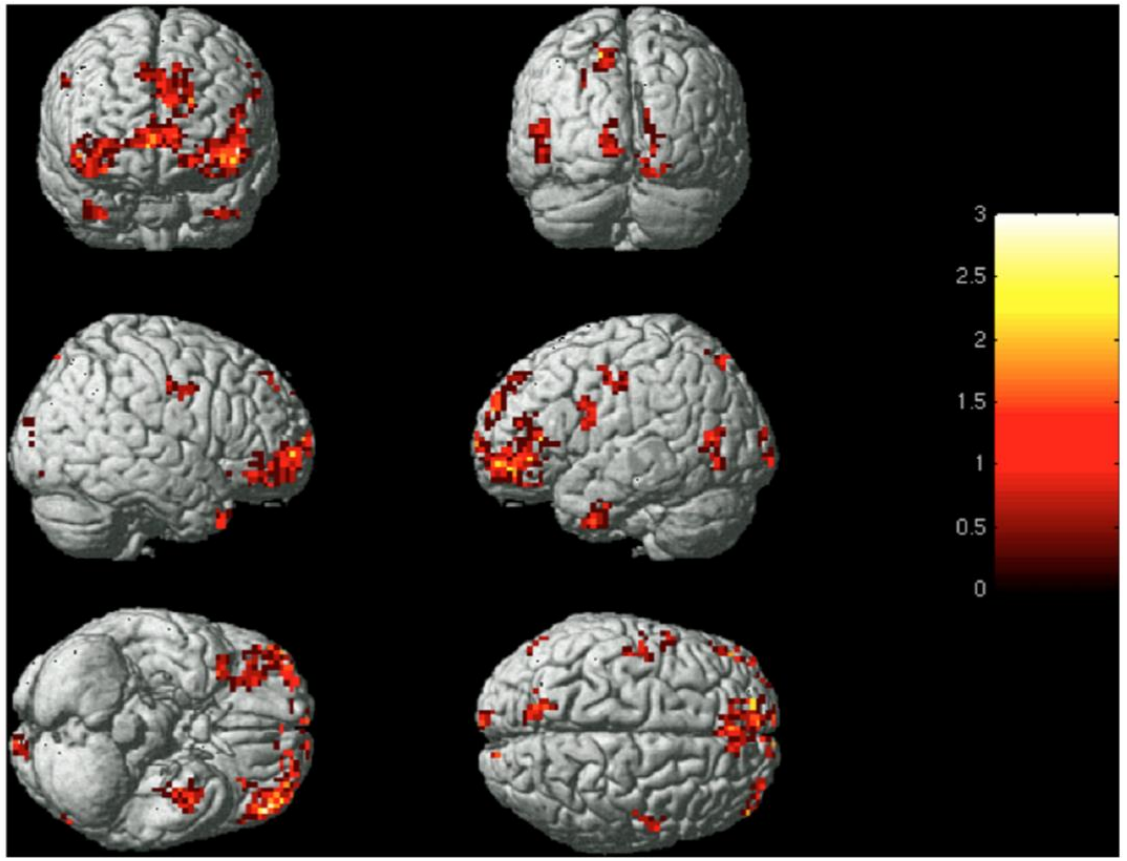


Figure 3. Group-level image of voxels used for within-subject classification. Images of the 240 most stable voxels across six presentations of emotion words were superimposed on one another, with resulting clusters of 25 or more voxels depicted. Color intensity reflects the number of participants for whom the voxel was among the 240 highest in stability.
doi:10.1371/journal.pone.0066032.g003

Berkley semantic map

Natural speech reveals the semantic maps that tile human cerebral cortex *by* Alexander G. Huth, Wendy A. de Heer, Thomas L. Griffiths, Frederic E. Theunissen & Jack L. Gallant (Nature, 2016)

extremetech.com/extreme/227498-uc-berkeley-team-built-a-semantic-atlas-of-the-human-brain

http://www.nature.com/nature/journal/v532/n7600/full/nature17637.html

From authors

“Our goal in this study was to map how the brain represents the meaning (or “semantic content”) of language.

Most earlier studies of language in the brain have used isolated words or sentences.

We used natural, narrative story stimuli because we wanted to map the full range of semantic concepts in a single study.

This made it possible for us to construct a semantic map for each individual, which shows which brain areas respond to words with similar meaning or semantic content.

Another aim of this study was to create a semantic atlas by combining data from multiple subjects, showing which brain areas represented similar information across subjects”

What's new there?

"We used natural narrative language rather than simple words presented in isolation.

We used cross-validation on a separate dat a set to test model prediction performance and generalization.

We used voxel-wise modeling to construct a separate semantic tuning curve for each voxel in each participant.

We used a probabilistic and generative model of areas tiling cortex (PrAGMATiC) to construct the semantic atlas."

What were the main conclusions of the study?

“The resulting maps show that semantic information is represented in rich patterns that are distributed across several broad regions of cortex.

Furthermore, each of these regions contains many distinct areas that are selective for particular types of semantic information, such as people, numbers, visual properties, or places.

We also found that these cortical maps are quite similar across people, even down to relatively small details.”

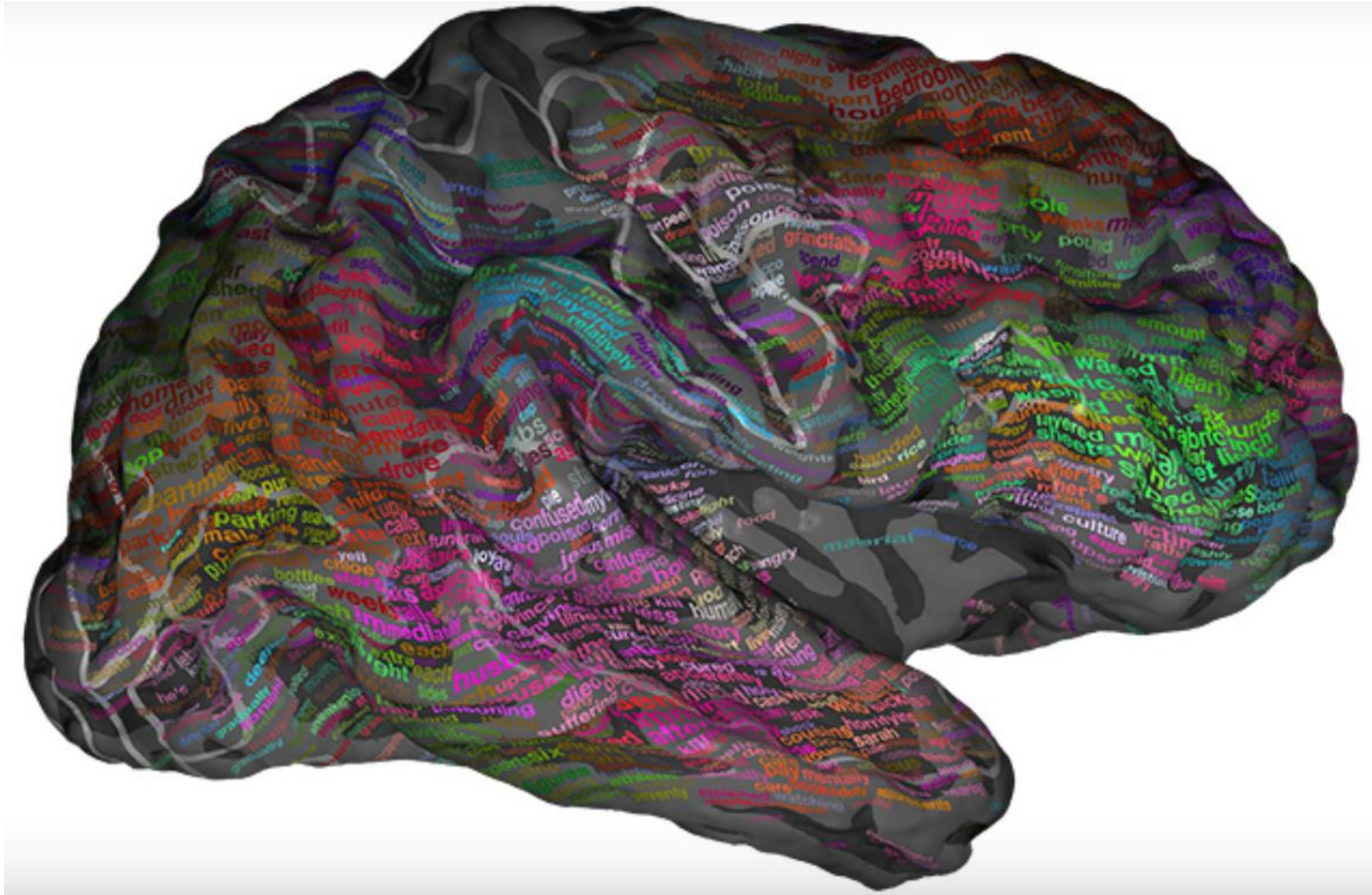
How does this study change the way that we think about language and the brain?

“These semantic maps give us, for the first time, a detailed map of how meaning is represented across the human cortex.

Rather than being limited to a few brain areas, we find that language engages very broad regions of the brain.

We also find that these representations are highly bilateral: responses in the right cerebral hemisphere are about as large and as varied as responses in the left hemisphere.

This challenges the current dogma (inherited from studies of language production, as opposed to language comprehension as studied here) holding that language involves only the left hemisphere.”



A known research solution

Standardized low resolution brain electromagnetic tomography

*s*LORETA

Exact low resolution brain electromagnetic tomography

*e*LORETA

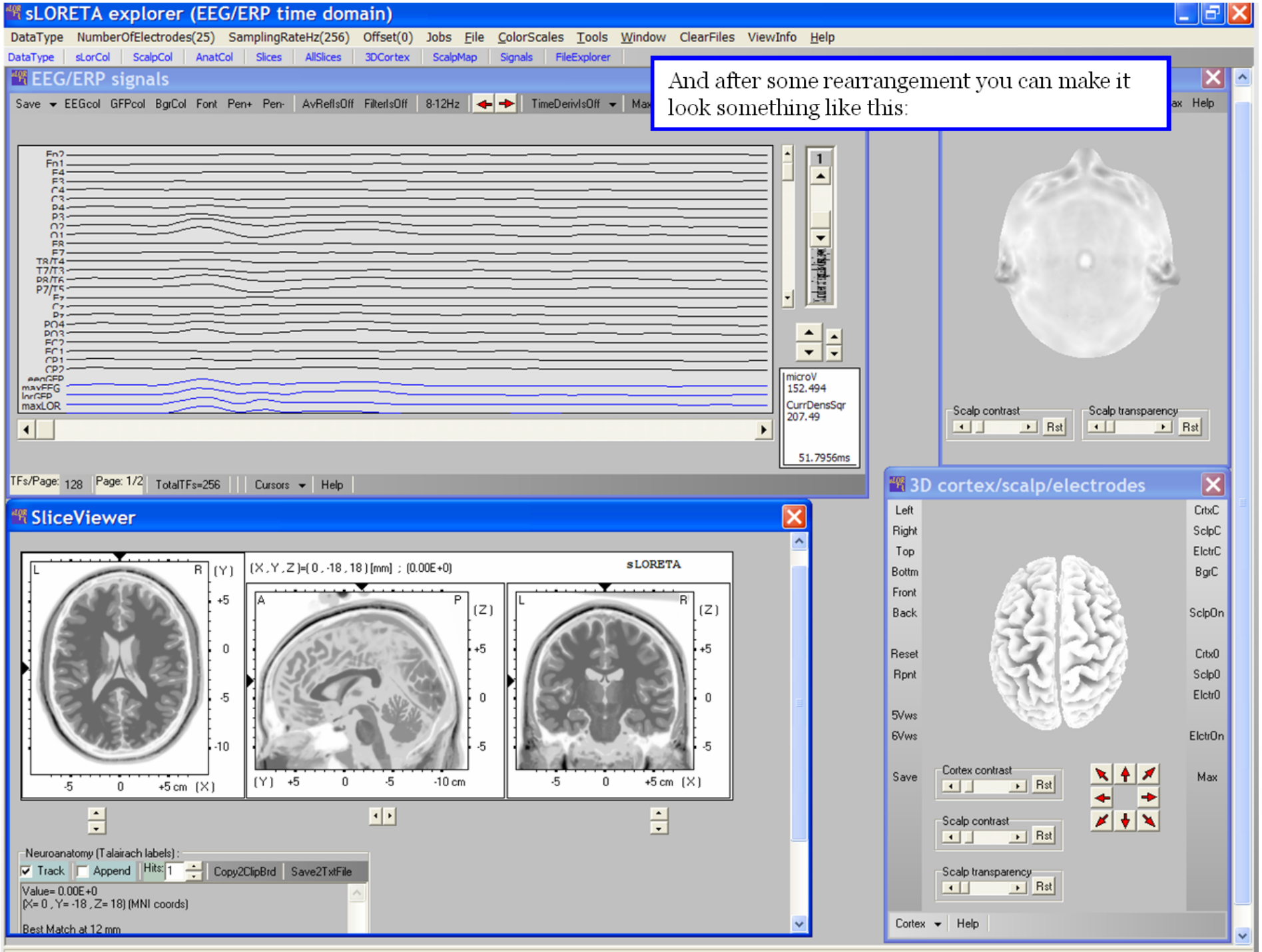
zero error, like it or not

Functional connectivity

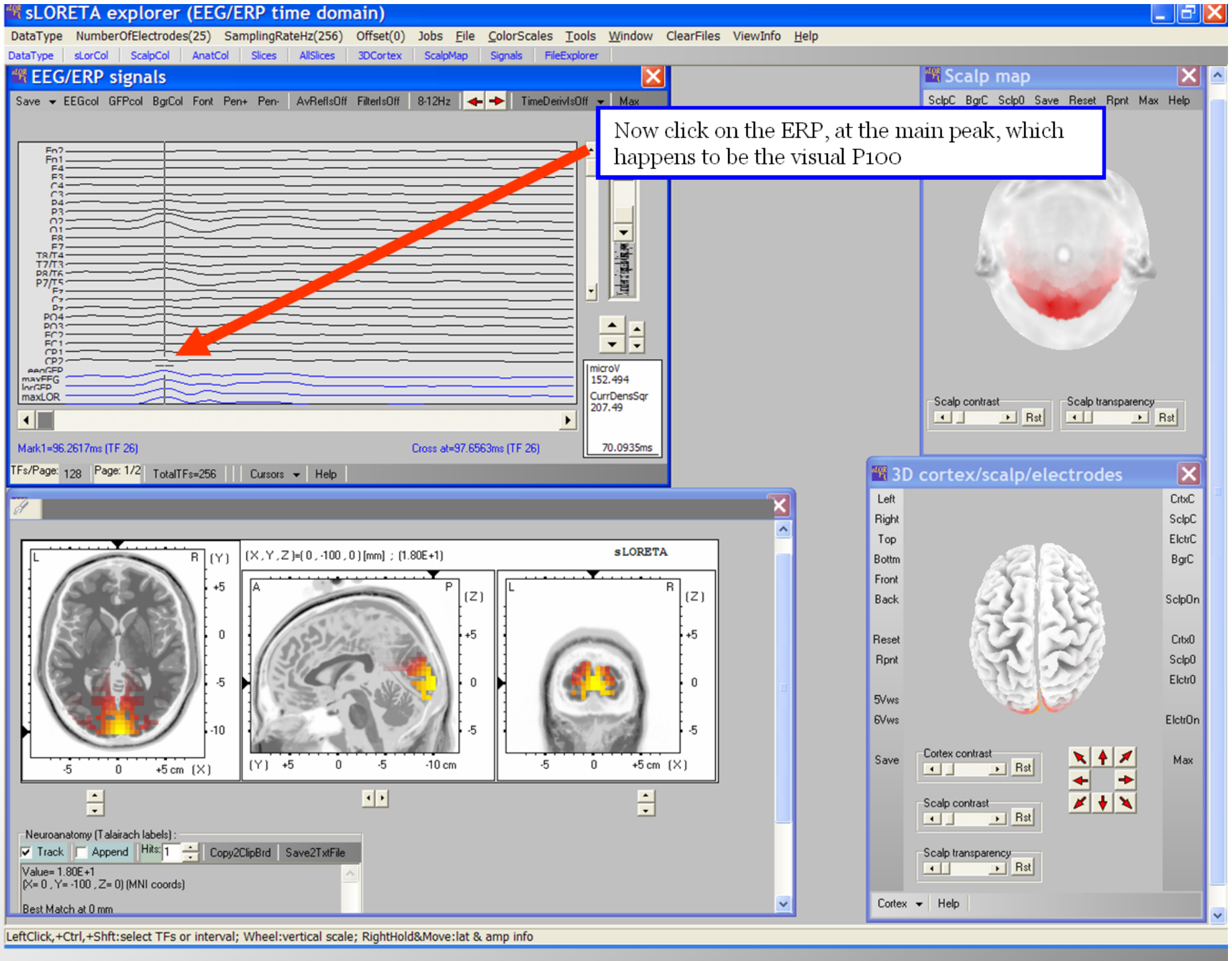
Lagged physiological coherence and phase synchronization, functional ICA (fICA), isolated effective coherence (iCoh)

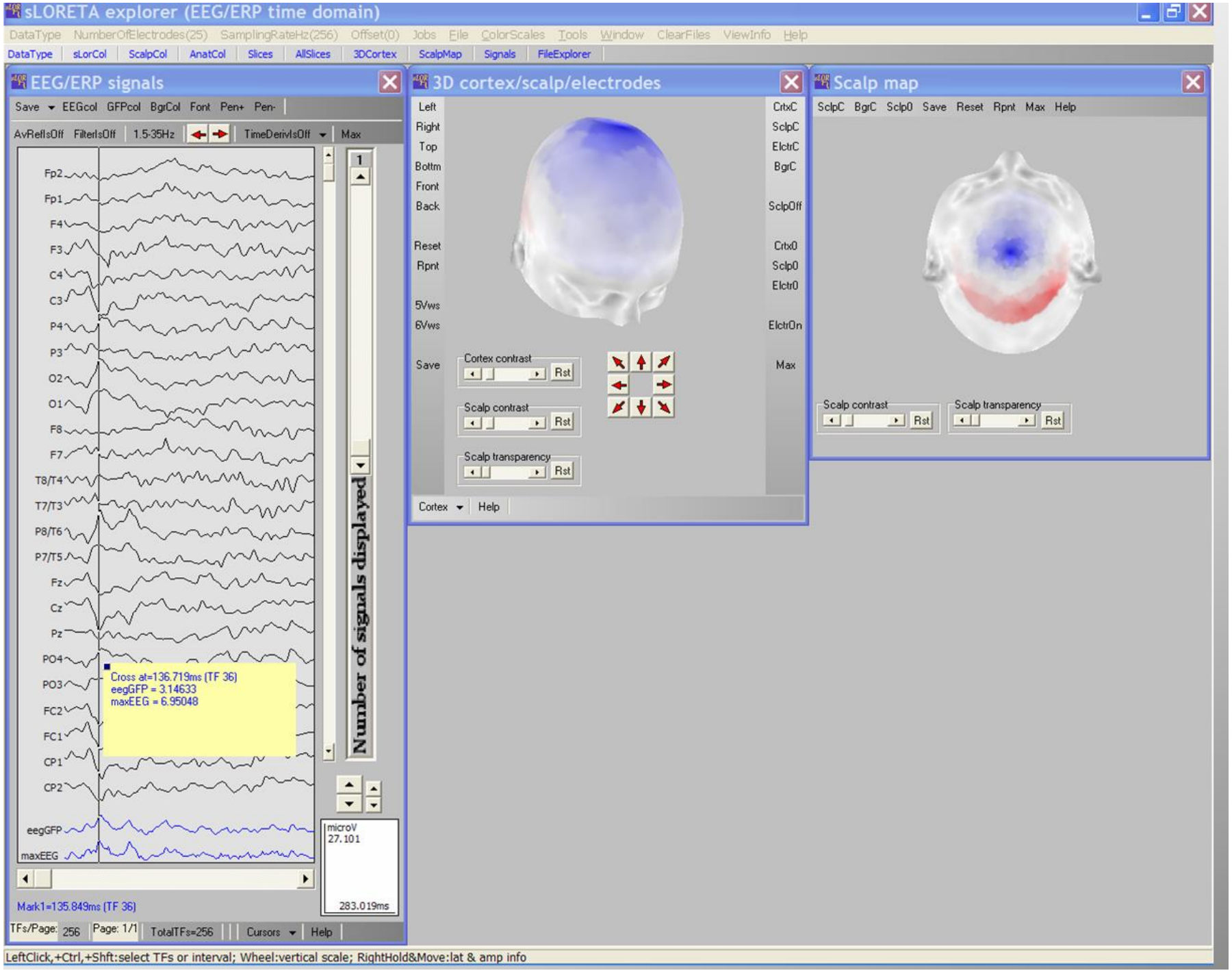
Roberto D. Pascual-Marqui
The KEY Institute for Brain-Mind Research
University Hospital of Psychiatry, Zurich Swiyzerland
p a s c u a l m @ key . u z h . c h
www.keyinst.uzh.ch/loreta.htm

Guest Professor at Department of Neuropsyquiatry
Kansai Medical University, Osaka, Japan



And after some rearrangement you can make it look something like this:


sLORETA explorer (EEG/ERP time domain)
DataType NumberOfElectrodes(25) SamplingRateHz(256) Offset(0) Jobs File ColorScales Tools Window ClearFiles ViewInfo Help
DataType sLorCol ScalpCol AnatCol Slices AllSlices 3DCortex ScalpMap Signals FileExplorer
EEG/ERP signals
Save EEGcol GFPcol BgrCol Font Pen+ Pen- AvRefIsOff FilterIsOff 8-12Hz TimeDerivIsOff Max
Now click on the ERP, at the main peak, which happens to be the visual P100
microV 152.494
CurrDensSqr 207.49
70.0935ms
Mark1=96.2617ms (TF 26)
Cross at=97.6563ms (TF 26)
TFs/Page: 128 Page: 1/2 TotalTFs=256 Cursors Help
Scalp map
SclpC BgrC Sclp0 Save Reset Rpnt Max Help
Scalp contrast Rst
Scalp transparency Rst
(X,Y,Z)=(0,-100,0) [mm] ; (1.80E+1)
sLORETA
Neuroanatomy (Talairach labels):
Track Append Hits: 1 Copy2ClipBrd Save2TxtFile
Value= 1.80E+1
(X= 0 , Y= -100 , Z= 0) (MNI coords)
Best Match at 0 mm
3D cortex/scalp/electrodes
Left Right Top Bottm Front Back Reset Rpnt 5Vws 6Vws Save
CrtxC SclpC ElctrC BgrC SclpOn Crtx0 Sclp0 Elctr0 ElctrOn Max
Cortex contrast Rst
Scalp contrast Rst
Scalp transparency Rst
Cortex Help
LeftClick,+Ctrl,+Shft:select TFs or interval; Wheel:vertical scale; RightHold&Move:lat & amp info


sLORETA explorer (EEG/ERP time domain)
DataType NumberOfElectrodes(25) SamplingRateHz(256) Offset(0) Jobs File ColorScales Tools Window ClearFiles ViewInfo Help
DataType sLorCol ScalpCol AnatCol Slices AllSlices 3DCortex ScalpMap Signals FileExplorer
EEG/ERP signals
Save EEGcol GFPcol BgrCol Font Pen+ Pen-
AvRefIsOff FilterIsOff 1.5-35Hz TimeDerivIsOff Max
Fp2
Fp1
F4
F3
C4
C3
P4
P3
O2
O1
F8
F7
T8/T4
T7/T3
P8/T6
P7/T5
Fz
Cz
Pz
PO4
PO3
FC2
FC1
CP1
CP2
eegGFP
maxEEG
Cross at=136.719ms (TF 36)
eegGFP = 3.14633
maxEEG = 6.95048
1
Number of signals displayed
microV
27.101
283.019ms
Mark1=135.849ms (TF 36)
TFs/Page: 256 Page: 1/1 TotalTFs=256 Cursors Help
3D cortex/scalp/electrodes
Left
Right
Top
Bottm
Front
Back
Reset
Rpnt
5Vws
6Vws
Save
CrtxC
SclpC
ElctrC
BgrC
SclpOff
Crtx0
Sclp0
Elctr0
ElctrOn
Max
Cortex contrast
Rst
Scalp contrast
Rst
Scalp transparency
Rst
Cortex Help
Scalp map
SclpC BgrC Sclp0 Save Reset Rpnt Max Help
Scalp contrast
Rst
Scalp transparency
Rst
LeftClick,+Ctrl,+Shft:select TFs or interval; Wheel:vertical scale; RightHold&Move:lat & amp info

Our research objective – MindDriver project: EEG-based real time tomography

Formulation of the research

1. Experimental research
   - Using eMotiv or any similar headset collect EEG data for different cognitive and emotional states
   - Can be done in both temporal and frequency domains
   - Collected data are interpreted using ML (as we did last spring) for two emotional modes
2. Theoretical research
   1. Create a forward model – from activated voxels to the boundary surface potentials
   2. Associate voxel with specific cognitive and emotional states (setting *f-labels*)
   3. Collect modeled data: voxels activation mapped to the surface potential
   4. Create the ML model mapping a given surface potentials onto *f-labels*

Experimental research

- BCI study
- The project covered mostly such areas as applications for educational use and recognition of emotions
- Among others, the following subjects were investigated:
  - ❖Emotiv Xavier SDK
  - ❖A BCI Learning Engine
  - ❖BCI based slide show presentation
  - ❖Emotional response to the educational material

Theoretical research under consideration

- Inverse problems solved by Pasqual-Marqui (sLoreta)
  - Issues: sensitivity to noise,
  - ill posed matrix,
  - undetermined equations
- Forward Monte Carlo neural activity simulation and construction of ML model

What is EEG mapping

- To establish correspondence between the EEG in discrete points on the head surface and the activation zones in the brain
- These zones are responsible for (presumably) corresponding to certain ideas, functions, emotions, etc.
- The zones are considered to be *voxels* as sections of a 3D brain model.

Our approach to investigate

- Generate activation voxels and calculate the potential on the surface
- Collect data to create a Machine Learning model: EEG to f-label
- Use of Monte-Carlo, Latin cube, etc.
- Avoid ill-posed matrix operations

Our approach - why

- Using EEG to identify what's "on your mind" is faster (real time) than MRI and fMRI used in the many published works.
- Opportunities to take an instantaneous electrical signals (EEG) from the scalp and map them onto the *f-labels* (emotions, words, etc.)

Forward model

- Neural network level
- Can be huge – really Big Data model
- Currently model of mice brains introduced with more than 30 million synapses (!)

Forward model: Method of Fundamental Solutions

The brain electrical field *u* can be modeled as an elliptical equation:

$$Lu = 0, \qquad (x, y, z) \in G,$$
$$u = g(x, y, z); \qquad (x, y, z) \in \partial G$$

where *G* is the domain (the brain);

$g(x, y, z)$ is the boundary conditions (the potentials on the surface of the brain).

MFS

Using the method of fundamental solutions means to construct solution's approximation, *u**(*x,y,z*), as the linear combination of the basis functions of the elliptical equation (functions of the distance between the source points and the collocation points on the boundary where the electrodes are located):

$$u^*(x,y,z) = \sum_{i=1}^{N} \alpha_i \varphi(r_i)$$

$$r_i = \|(x,y,z) - (sx_i, sy_i, sz_i)\|$$

where (sx_i, sy_i, sz_i) is the location of the source (the center of the voxel).

Basis functions

In the 3D case the basis function

$$\varphi(r_i) = 1/r_i$$

The MFS representation is equivalent to the model used in LORETA, that is a set of local electrical sources.

Inverse problem approach

- The Monte-Carlo method is suggested because it is supposed to avoid the instability of the inverse method that leads to the operations with the ill-imposed matrices.
- The algorithm is a sequence of the direct solutions – from the sources to the boundary points.
- The values of the potentials in the sources are selected randomly.

Algorithm

- The solutions are registered at the boundary points.
- The solutions can be presented as the look-up table.
- This Big Dataset is used in the real time mapping of the electrode potentials to the voxels intensity. The measured vector of scalp potentials is compared with the closest match using some norm (the max(.) norm may work better).

OR (AND), which is what we are going to study

- A Machine Learning model should be created to map the scalp EEG potentials onto the f-labels
- f-labels are functions (such as emotions, words, etc.) associated with specific voxels

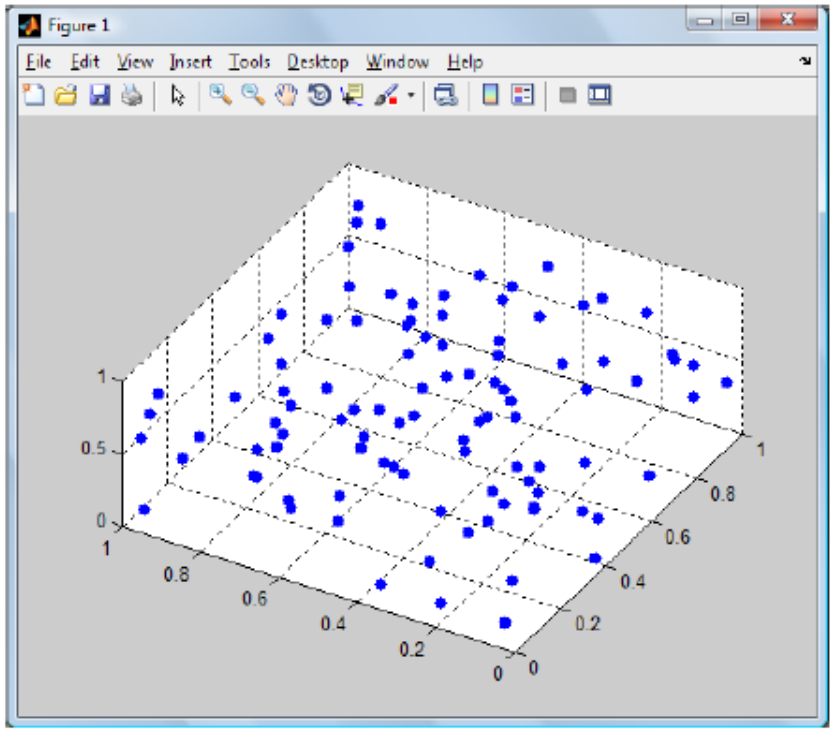
Latin Hypercube

- The Monte Carlo method is based on propagating probability distributions.
- Given a region in design space, we can assign a uniform distribution to the region and sample points.
- It is likely, though, that some regions will be poorly sampled.
- For example, in 5-dimensional space, with 32 sample points, the chance that all *n*- dimensional quadrants (orthants) will be occupied is (31/32)(30/32)(1/32) …(1/32) =$1.8e^{-13}$.

Latin Hypercube sampling

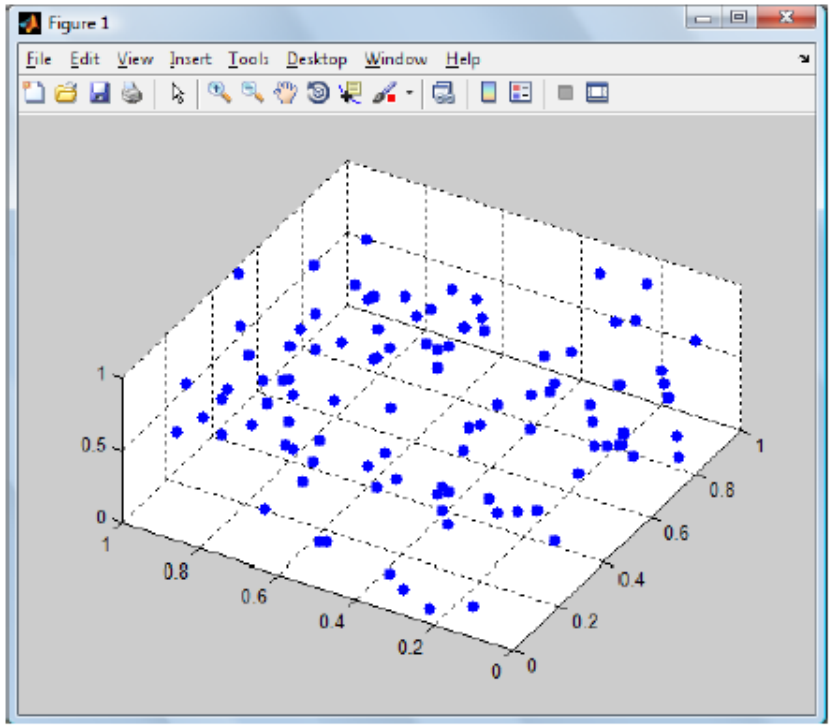
- To scan the parametric space deterministically, a prohibitively high number of calculations is needed.
- Monte Carlo simulation: the chance is that the intervals are sampled with higher density because the same random sample can contribute to more than one parameter's sampling.
- It can be a waste of simulation runs if the clusters are formed in less important parts of parametric space
- Latin Hypercube, orthogonal sampling, uniform space filing methods and others can be used.
- Applied to both sensitivity analysis and parametric interval control.
- Each variable range divided into *n* equal probability intervals.

MATLAB models of LHC



1st run

```
scatter3(XX(1,:),XX(2,:),XX(3,:),
'filled'),view(-60,60)
```



2nd run

```
scatter3(XX(1,:),XX(2,:),XX(3,:),
'filled'),view(-60,60)
```

Forward modeling for BCI research

- Create a forward model – from activated voxels to the boundary surface potentials
- Associate voxel with specific cognitive and emotional states (setting *f-labels*)
- Collect modeled data: voxels activation mapped to the surface potential
- Create the ML model mapping a given surface potentials onto *f-labels*

Static and dynamic models

Assumption: the potentials on the surface are resulted from a dynamic process of the wave of neuron activations.

Each neuron's process is defined as a set of dynamic equations such as

$$\frac{dy}{dt} = F(x, y, t)$$

where $x(t), y(t)$ are input and output time t -dependant functions; $F()$ is a non-linear and nonstationary function representing the dynamic property as well.

The input of a neuron is under influence of many other neurons.

For example, a simplified model can be presented as

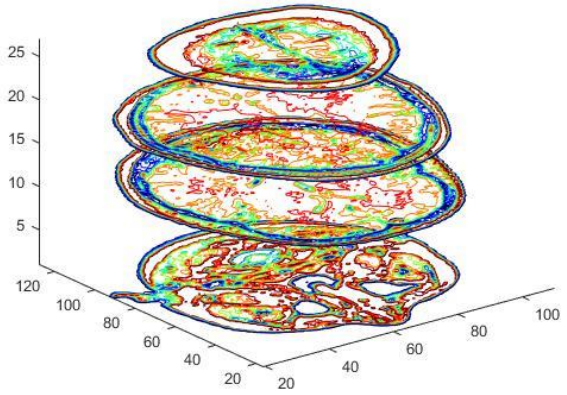
$$\frac{dy}{dt} = k\sigma\left(\sum \alpha_i \, y_i\right)$$

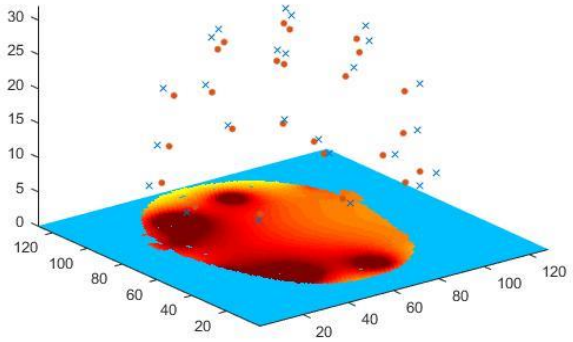
where k is a time constant and $\sigma(.)$ is a nonlinear sigmoid function applied to a sum of the outputs y_i of the connected neurons.

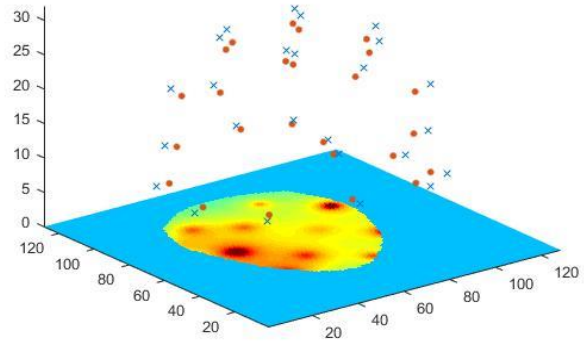
Dynamic, random and fractal models

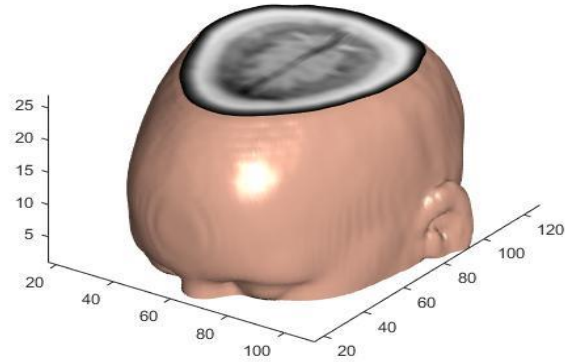
- Activation of a set of seed neurons leads to propagation of the wave of electrical potential chain until it reaches the surface.
- NEURON Simulator (www.neuron.yale.edu)
- Such dynamic models allow for performing analysis in the frequency domain allowing to deal with a limited number of parameters (alpha, delta, theta, etc.).
- Dynamic modeling should include also the creation of chain of the neuron activations. It is suggested that a *fractal* (self-similarity) model should be considered. The fractal dimension can be used to characterize time series of potentials instead of analysis in the frequency domain.

Experimental results

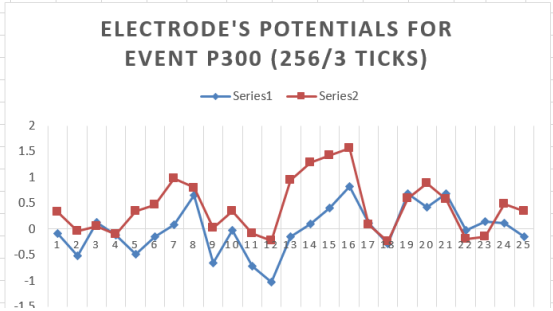






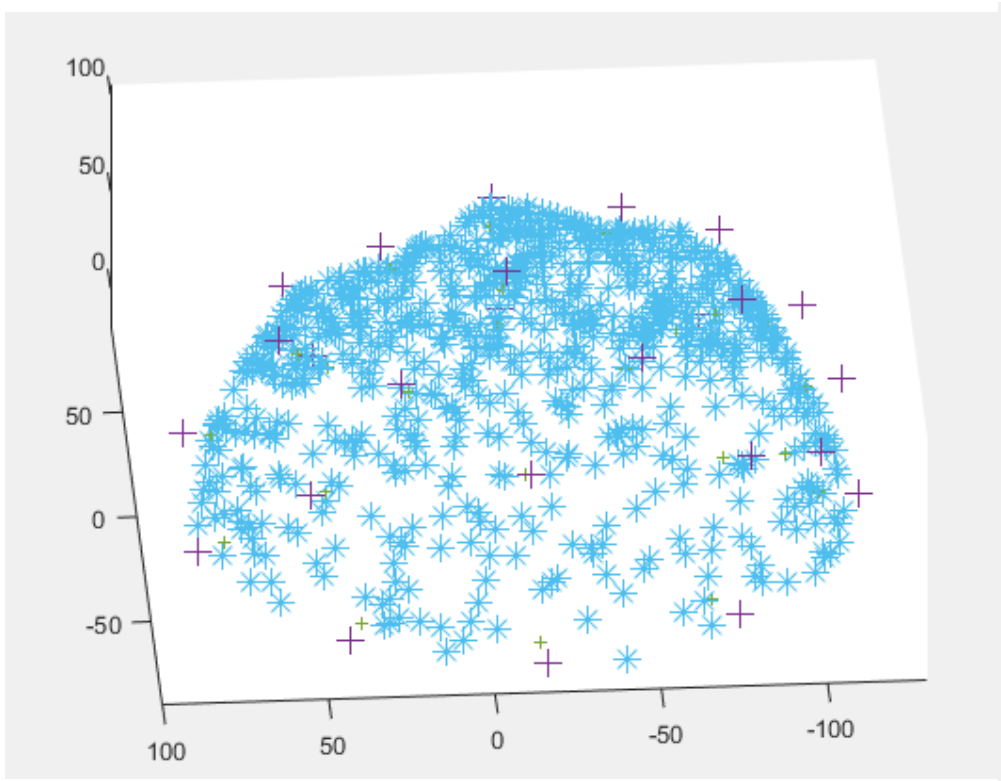


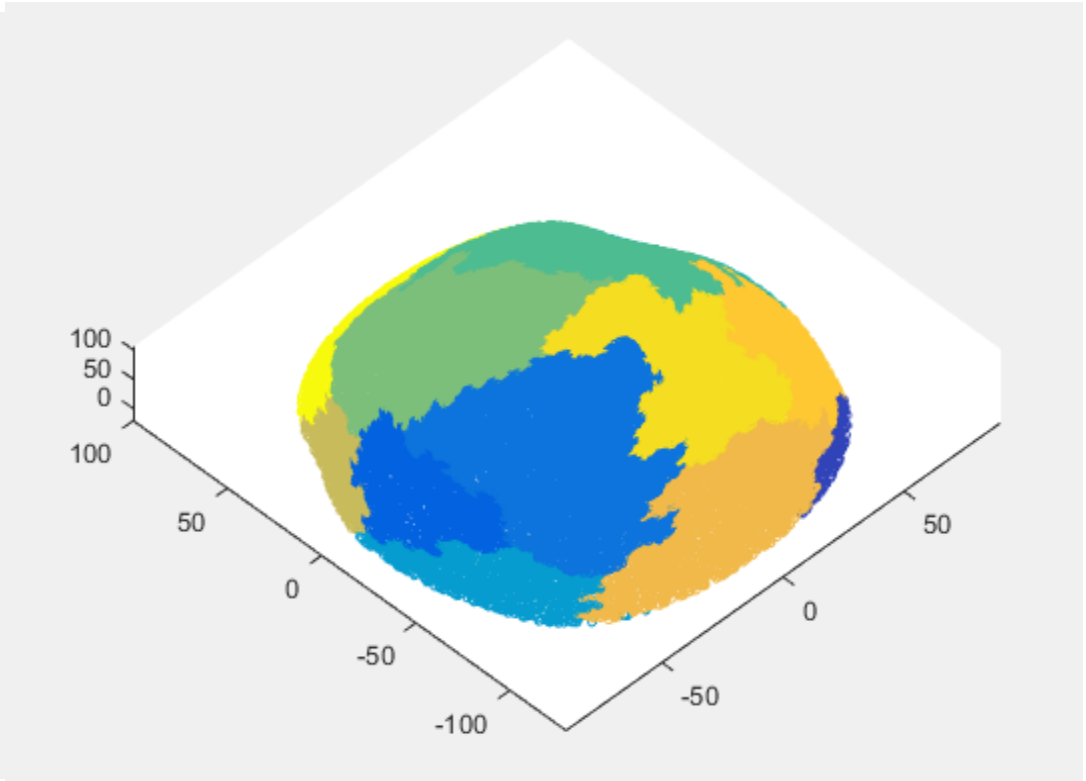
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC
-0.08	-0.52	0.1	-0.12	-0.49	-0.14	0.1	0.7	-0.65	-0.03	-0.72	-1.02	-0.15	0.1	0.4	0.8	0.1	-0.28	0.7	0.4	0.7	-0.02	0.1	0.1	-0.15		Event Related Potential: Grey (blue)		
0.33	-0.04	0.1	-0.1	0.35	0.47	1	0.8	0.02	0.34	-0.09	-0.22	0.95	1.3	1.4	1.6	0.1	-0.24	0.6	0.9	0.6	-0.19	-0.2	0.5	0.34		Event Related potential: Flower (red)		



Input.txt contains 10000 combinations of source values at 1000 locations (clusters) on the cortex; Output.txt has the corresponding potentials expected at the electrodes.

Active areas in the brain (f-labels)



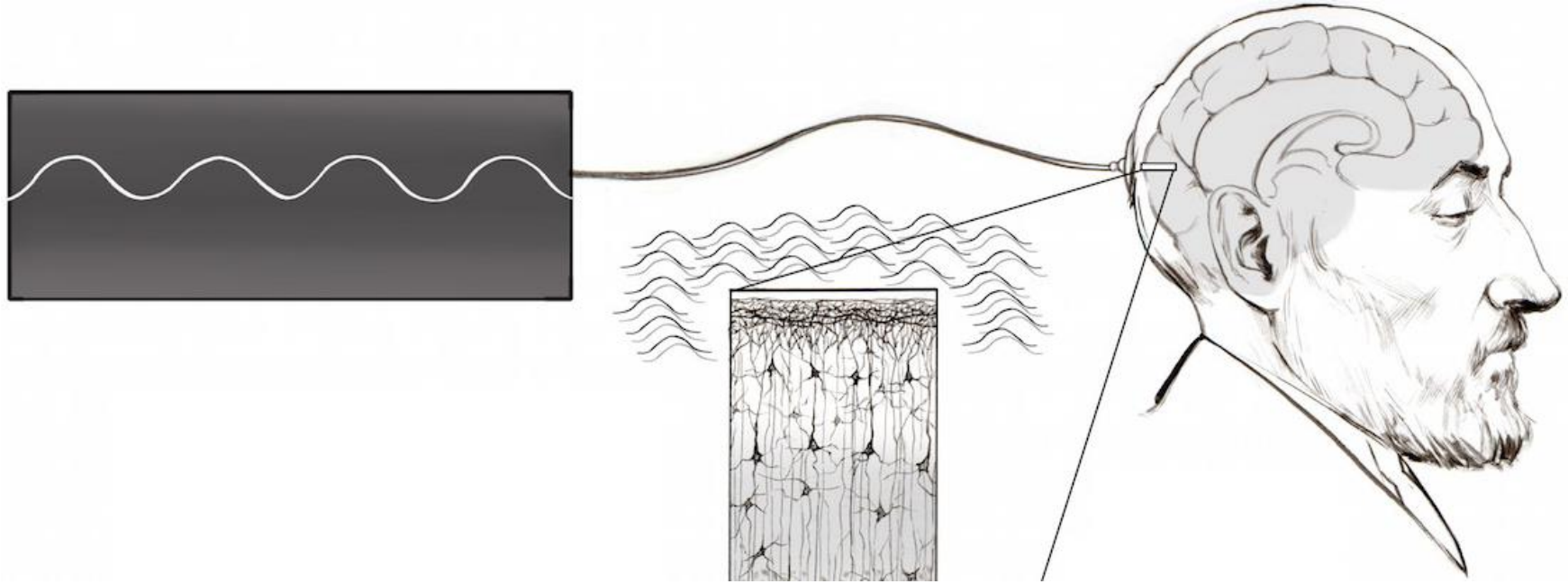


Big Data: Machine Learning

- During the last fall (F 2016) semester, two term project research graduate groups continued working on this project.
- Below are some results presented in their report.

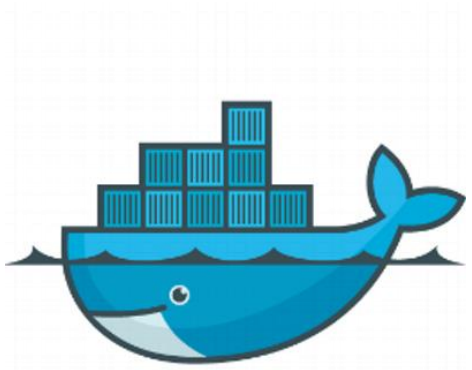
Problem

- Develop and train a machine learning model which can correctly associate signals from activated source points within the brain to EEG sensors
- Develop a machine learning model which can then correctly associate EEG activity with the correct sources within the brain



Tools

- TensorFlow
  - Open source Python machine learning library by Google
  - Meant for computation of complex data flow graphs
- Docker
  - Lightweight virtualization application
  - Maintains a library of system images to enable quick implementation of software with all necessary dependencies
- Jupyter Notebooks
  - Used for data science applications
  - Compartmentalizes portions of code into cells that can be run separately



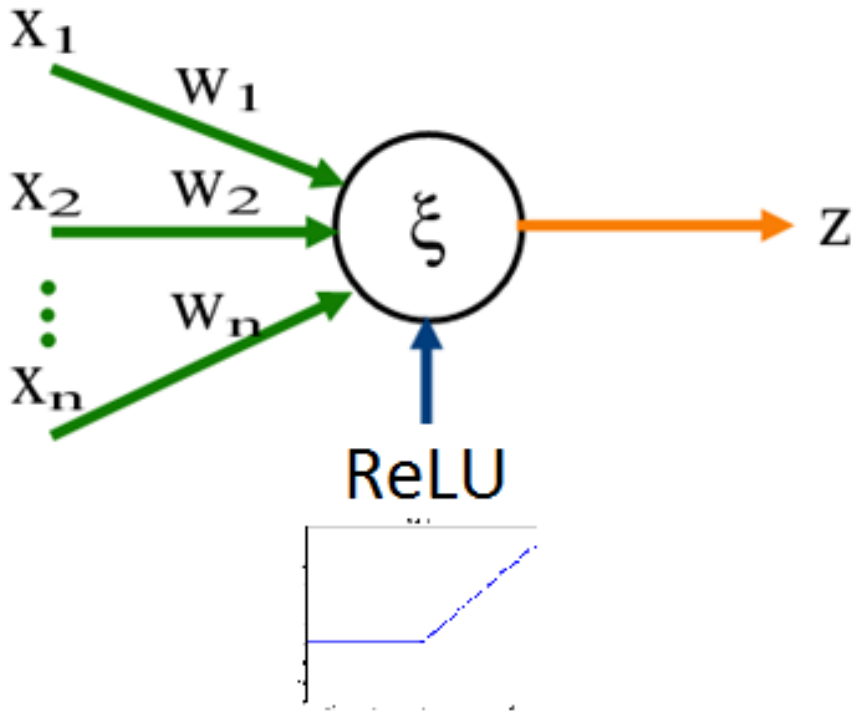
Training Data

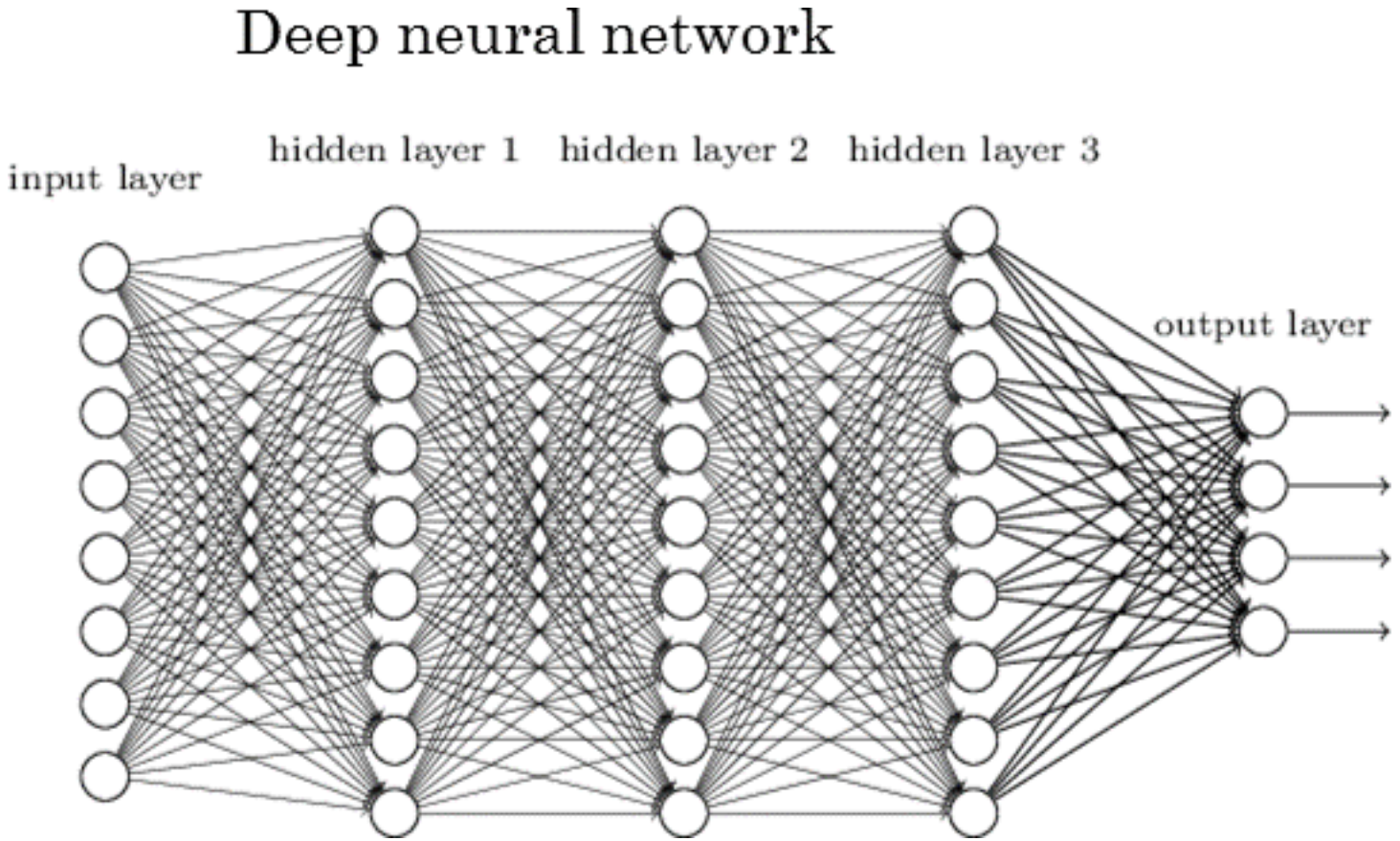
- A set of 2000 1x625 arrays consisting of source potential signals received by each electrode, with random sources activated
- A set of 2000 1x25 arrays with a 1 indicating the electrode receiving the strongest total signal
- The sets are coupled and fed to the neural network

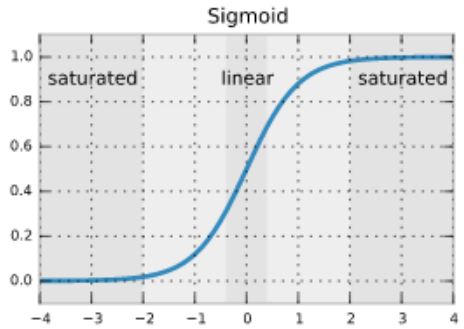
Electrode	Source 1	Source 2	Source 3	Source 4	Source 5	Source 6
1	0.111956	0	0.016095	0.010684	0.008244	0.006794
2	0.018517	0	0.010609	0.016171	0.006831	0.00816
3	0.016525	0	0.120093	0.010647	0.015071	0.007826
4	0.010848	0	0.010687	0.123195	0.007903	0.014847
5	0.0082	0	0.014492	0.0078	0.106447	0.007956
6	0.006772	0	0.007747	0.014232	0.007962	0.107227
7	0.005813	0	0.007753	0.006202	0.014512	0.0076
8	0.005307	0	0.006126	0.007722	0.007472	0.01464
9	0.005237	0	0.00607	0.00562	0.008408	0.007011

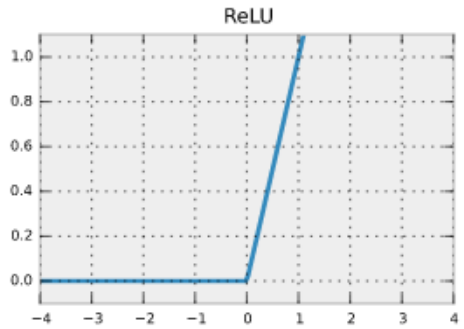
Electrode	1	2	3	4	5	6	7	8	9
Sum of Sources	0.153773	0.060288	0.170162	0.16748	0.144894	0.143941	0.041881	0.041266	0.032348
One Hot	0	0	1	0	0	0	0	0	0

Neural Networks with Rectified Linear Unit (ReLU)









Neural Network Model

- 1 input layer
- 3 hidden layers
- 1 output layer
- 500 nodes per hidden layer

```
def reverse_network_model(data):

    hidden_1_layer = {'weights': tf.Variable(tf.random_normal([625, n_nodes_hl1])),
                      'biases':tf.Variable(tf.random_normal([n_nodes_hl1]))}

    hidden_2_layer = {'weights': tf.Variable(tf.random_normal([n_nodes_hl1,n_nodes_hl2])),
                      'biases':tf.Variable(tf.random_normal([n_nodes_hl2]))}

    hidden_3_layer = {'weights': tf.Variable(tf.random_normal([n_nodes_hl2, n_nodes_hl3])),
                      'biases':tf.Variable(tf.random_normal([n_nodes_hl3]))}

    output_layer = {'weights': tf.Variable(tf.random_normal([n_nodes_hl3, n_classes])),
                      'biases':tf.Variable(tf.random_normal([n_classes]))}

    l1 = tf.add(tf.matmul(data, hidden_1_layer['weights']), hidden_1_layer['biases'])
    l1 = tf.nn.relu(l1)

    l2 = tf.add(tf.matmul(l1, hidden_2_layer['weights']), hidden_2_layer['biases'])
    l2 = tf.nn.relu(l2)

    l3 = tf.add(tf.matmul(l2, hidden_3_layer['weights']), hidden_3_layer['biases'])
    l3 = tf.nn.relu(l3)

    output = tf.matmul(l3, output_layer['weights']) + output_layer['biases']

    return output
```

Training

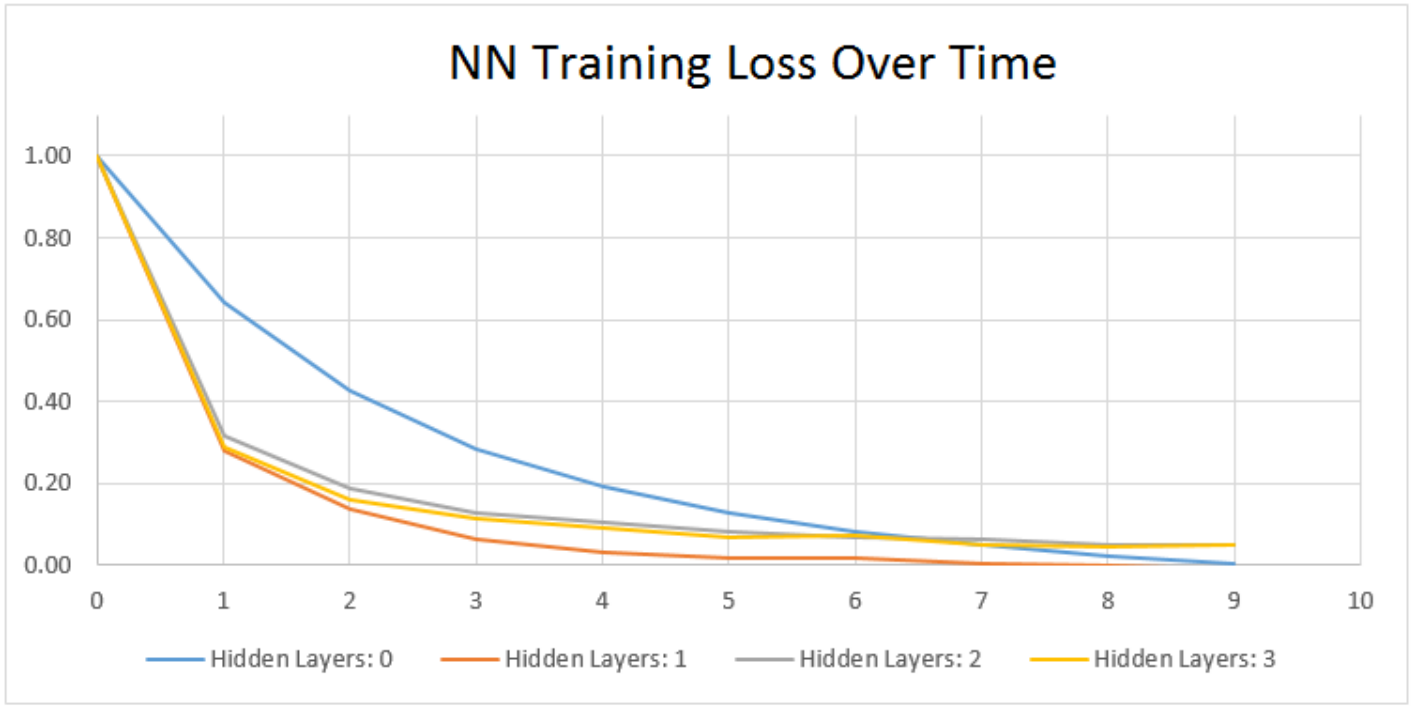
- Passes data from the training set to the model for training

```
def train_neural_network(x):
    prediction = reverse_network_model(x)

    # The model will use cross-entropy with soft-max
    # this defines the cost function, but doesn't do anything yet
    cost = tf.reduce_mean( tf.nn.softmax_cross_entropy_with_logits(prediction, y) )

    # Optimize the model, using a default learning_rate=0.001
    # This defines the optimizer, doesn't do anything yet
    optimizer = tf.train.AdamOptimizer().minimize(cost)

    # total cycles of feedforward + backpropagation
    hm_epochs = 10

    with tf.Session() as sess:
        sess.run(tf.initialize_all_variables())

        for epoch in range(hm_epochs):
            epoch_loss = 0
            # get each batch of data, then run through the model
            for i in range(training_data.shape[0]):
                #x is the raw data, y is the labels
                #epoch_x, epoch_y = [training_data[i]]
                epoch_x, epoch_y = training_set[i]
                #run the optimizer with cost
                _, c = sess.run([optimizer, cost], feed_dict= {x: np.reshape(epoch_x,(-1, 62
5)), y: epoch_y})
                epoch_loss += c
            print('Epoch ', epoch, ' completed out of ', hm_epochs, ' loss:', epoch_loss)

        #This compares the predicted results and the expected results
        #correct results are closer to
        correct = tf.equal(tf.argmax(prediction,1), tf.argmax(y,1))
        accuracy = tf.reduce_mean(tf.cast(correct, 'float'))
        print('Accuracy: ', accuracy.eval({x:training_data,
                                           y:training_labels}))

train_neural_network(x)
```

Results

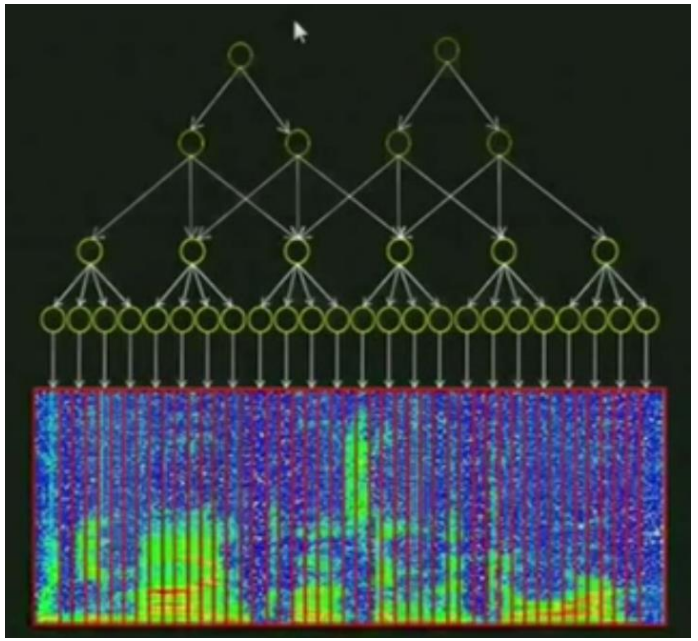
- An identical dataset was tested against neural networks consisting of 0, 1, 2, and 3 hidden layers of 500 nodes

# Hidden Layers	Accuracy
0	77.9%
1	91.9%
2	93.9%
3	91.8%



Unsupervised machine learning method

Deep Learning as a DBN



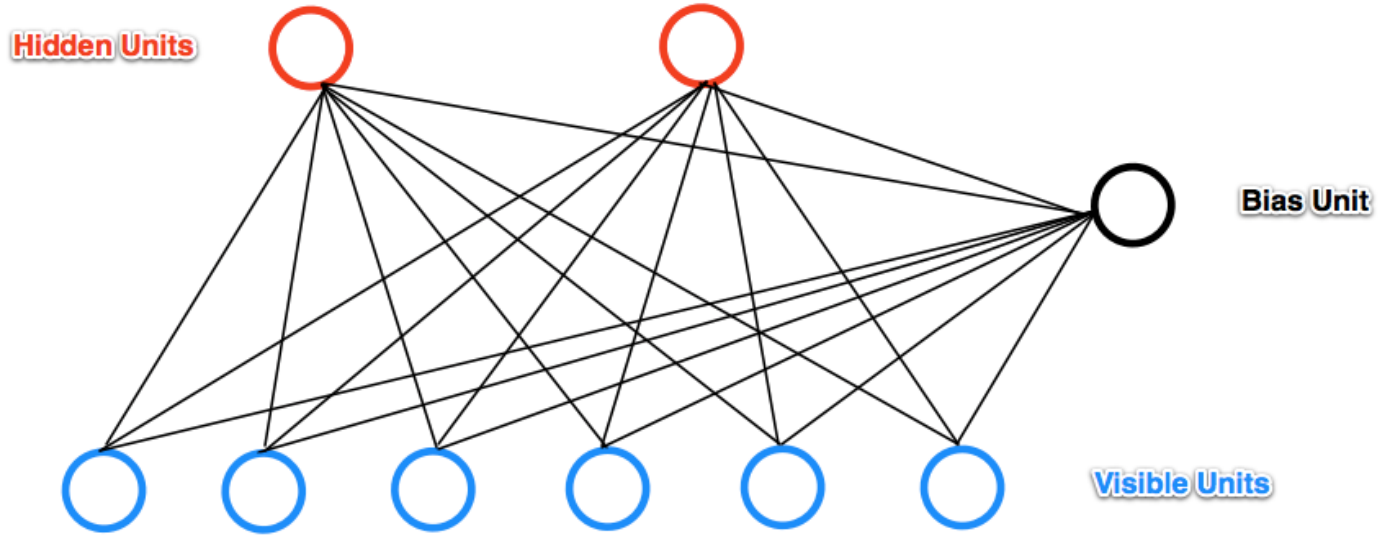
- A fragment of a Deep Belief Network
- Unsupervised learning
- Spectrogram as an input (*visual nodes*)
- Stacked layers
- Supervised layer as a final one
- Parallel solutions at each stage (Map-Reduce)
- Backpropagation at the last stage

Deep Learning classification

- Deep networks for unsupervised or generative learning, which are intended to capture high-order correlation of the observed or visible data for pattern analysis or synthesis purposes when no information about target class labels is available.
- Deep networks for supervised learning, which are intended to directly provide discriminative power for pattern classification purposes, often by characterizing the posterior distributions of classes conditioned on the visible data. Target label data are always available in direct or indirect forms for such supervised learning. They are also called discriminative deep networks.
- Hybrid deep networks, where the goal is discrimination which is assisted, often in a significant way, with the outcomes of generative or unsupervised deep networks.

Restricted Boltzmann Machine

- The most widely used hybrid deep architecture
- A special type of Markov random field that has one layer of (typically Bernoulli) stochastic hidden units and one layer of (typically Bernoulli or Gaussian) stochastic visible or observable units



RBM: theory and method

The joint distribution $p(\mathbf{v},\mathbf{h};\vartheta)$ over the visible units v and hidden units h, given the model parameters ϑ, is defined in terms of an energy function $E(v,h;\vartheta)$ of

$$p(\mathbf{v},\mathbf{h};\theta) = \frac{\exp(-E(\mathbf{v},\mathbf{h};\theta))}{Z}, \qquad Z = \sum_{\mathbf{v}}\sum_{\mathbf{h}} \exp(-E(\mathbf{v},\mathbf{h};\theta))$$

The marginal probability that the model assigns to a visible vector v is

$$p(\mathbf{v};\theta) = \frac{\sum_{\mathbf{h}} \exp(-E(\mathbf{v},\mathbf{h};\theta))}{Z}$$

The energy function is defined as

$$E(\mathbf{v},\mathbf{h};\theta) = -\sum_{i=1}^{I}\sum_{j=1}^{J} w_{ij}v_i h_j - \sum_{i=1}^{I} b_i v_i - \sum_{j=1}^{J} a_j h_j.$$

The conditional probabilities can be efficiently calculated as

$$p(h_j = 1|\mathbf{v};\theta) = \sigma\left(\sum_{i=1}^{I} w_{ij}v_i + a_j\right),$$

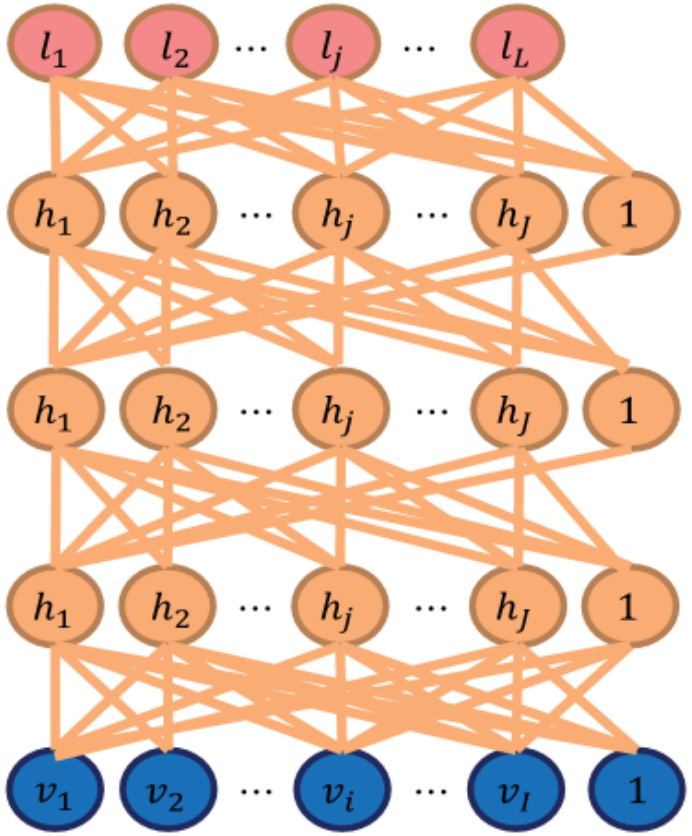
$$p(v_i = 1|\mathbf{h};\theta) = \sigma\left(\sum_{j=1}^{J} w_{ij}h_j + b_i\right),$$

where $\sigma(x) = 1/(1+\exp(-x))$.

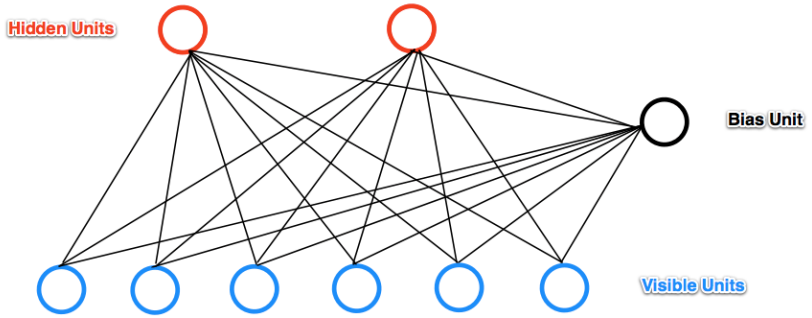
Deep Learning: Stacking the RBM's

- Stacking a number of the RBMs learned layer by layer from bottom up gives rise to a DBN.
- The stacking procedure is as follows. After learning a RBM with binary features such as spectral bins indexed for different electrodes, we treat the activation probabilities of its hidden units as the data for training the Bernoulli RBM one layer up.
- The activation probabilities of the second layer Bernoulli-Bernoulli RBM are then used as the visible data input for the third-layer Bernoulli-Bernoulli RBM, and so on.
- It has been shown that the stacking procedure improves a variational lower bound on the likelihood of the training data under the composite model.
- That is, this greedy procedure achieves approximate maximum likelihood learning.
- This learning procedure is unsupervised and requires no class label.

The Deep Belief Network – Deep Neural Network (DBN-DNN) architecture



Training example



Six visible nodes (V1..V6): theta rhythm presence (1) and otherwise (0) at different electrode locations

Two hidden nodes to differentiate unknown emotional states

Training samples: A(1,1,1,0,0,0), B(1,0,1,0,0,0), C(1,1,1,0,0,0), D(0,1,1,1,0,0), E(0,0,1,1,1,0), F(0,0,1,1,1,0)

The network learned the following weights:

	Bias Unit	Hidden 1	Hidden 2
Bias Unit	-0.08257658	-0.19041546	1.57007782
V1	-0.82602559	-7.08986885	4.96606654
V2	-1.84023877	-5.18354129	2.27197472
V3	3.92321075	2.51720193	4.11061383
V4	0.10316995	6.74833901	-4.00505343
V5	-0.97646029	3.25474524	-5.59606865
V6	-4.44685751	-2.81563804	-2.91540988

In the testing mode, a sample (0,0,0,1,1,) is tested. It turns Hidden 1 on and Hidden 2 off. Interpretation?

Big Data Brain model: What is the best choice of a platform

- All of this accessing and processing of data can be done entirely within MATLAB, no matter where the data might be stored including SQL/NoSQL databases, Spark™, and/or Hadoop®.
- There are also multiple choices when it comes to the Deep Machine Learning platform – Tensor Flow, the R system, MATLAB package and possibly more.

MATLAB: Big Data and tall arrays – a plausible choice

Advantages of MATLAB platform:

- Work with arrays of any size `tall` tables
- Deferred evaluations
- Looks like working with normal arrays in memory
- Evaluation occurs when `gather` command called
- Used datastore: `ds = datastore(eeg.csv)`
- Create a tall table from datastore: `eeg_tall = tall(ds)`
- Strong graphics support
- Vast mathematical support

In conclusion…

- Big Data models and Machine Learning methods are the natural choice in study of brain activities and human cognitive functions
- EEG tomography opens an opportunity to follow human emotions and thoughts real time
- The Method of fundamental solutions shows some promise in achieving this goal