Aalborg University, Biomedical Engineering and Informatics,
Spring Semester, 2012

Master Thesis

"Classification of movements of the rat based on intra-cortical signals using artificial neural network and support vector machine"

Martina Corazzol

AALBORG UNIVERSITY

Department of Health Science & Technology
Frederik Bajers Vej 7D2
9220 Aalborg
Telephone (+45) 9940 9940
Fax (+45) 9815 4008
http://www.hst.aau.dk

Title:
"Classification of movements of the rat based on intra-cortical signals using artificial neural network and support vector machine"

Project period:
Master Thesis, Spring 2012

Project group:
Group 12gr1085

Members:
Martina Corazzol

Supervisor:
Winnie Jensen

Assistant Supervisor:
Sofyan Hammad Himaidan Hammad

No. printed Copies: 3

No. of Pages: 75

No. of Appendix Pages: 27

Total no. of pages: 102

Completed: 15th of October 2012

Synopsis:

Abstract
The main goal of an intra-cortical brain computer interface (BCI) is to restore the lost functionalities in disabled patients suffering from severely impaired movements. A BCI has the advantage to create a direct communication pathway between the brain and an external device for restoring disabilities. This is possible by decoding signals from the primary motor cortex and translating them into commands for a prosthetic device. The project aim was to develop a decoding method based on a rat model. Previously recorded data and an already develop preprocessing method were used. The experimental design was developed starting from intra-cortical (IC) signal recorded in the rat primary motor cortex (M1). The data preprocessing included denoising with wavelet technique, spike detection, and feature extraction. After the firing rates of intra-cortical neurons were extracted, artificial neural network (ANN) and support vector machine (SVM) were applied to classify the rat movements into two possible classes, *Hit* or *No Hit*. The misclassification error rates obtained from denoised and not denoised data were statistically different ($p<0.05$), proving the efficiency of the denoising technique. ANN and SVM provided comparable misclassification errors, ranging between 14% and 39%.

Preface

This report has been composed by Martina Corazzol, during the 4th semester of the Master of Science in Biomedical Engineering and Informatics with speciality in Medical Systems at the Institute for Health Science and Technology at Aalborg University, Denmark.

This project is inserted in an ongoing study about decoding algorithm for characterizing and predicting primary motor cortex (M1) responses during a behavioural task. Therefore, the data recorded during the previous study were used in this project to classify the movements of the rat by using artificial neural network and support vector machine.

The reference style used in the report is according to the Harvard method [Last name, Year]. The references are indicated before and after a full stop. If a reference is indicated before a full stop it refers only to the sentence, whereas if it stands after a full stop it refers to the entire section. The references listed in succession are arranged by the year starting with the oldest one. Figures and tables are numbered with reference to the chapter e.g. figure 1 in chapter 2 is "Figure 2.1". The captions are set below the figures or tables.

I would like to thank my supervisor Winnie Jensen for the technical assistance during the project. Moreover, I would like to thank my co-supervisor Sofyan Hammad for the data he provides me and for the assistance during the data processing.

Martina Corazzol

Acronyms

BCI	Brain Computer Interface
BMI	Brain Machine Interface
CNS	Central Nervous System
ANN	Artificial Neural Network
SVM	Support Vector Machine
M1	Primary Motor Cortex
S1/S2	Somatosensory Area 1 and 2
PMA	Premotor Area, d dorsal and v ventral
SMP	Supplementary Motor Area
RFA	Rostral Forelimb Area
IC	Intra-Cortical Signal
CM	Corticomotor neurons
fMRI	Functional Magnetic Resonance Imaging
fNIRS	Functional Near-Infrared Spectroscopy
MEG	Magnetoencephalogram
EEG	Electroencephalogram
ECoG	Electrocorticogram
LFP	Local Field Potential
SUA	Single Unit Activity
MUA	Multi Units Activity

Table Of Contents

Table Of Contents 7

Chapter 1 Introduction 11
1.1 Initiating problem formulation 12

I Problem Analysis 13

Chapter 2 Elements of a BCI 15
2.1 Principal components of a BCI 15
2.2 The motor cortex 17
2.2.1 The primary motor cortex 18
2.2.2 The premotor cortex 18
2.2.3 The supplementary motor area 19
2.2.4 Role of the motor cortex in voluntary movement 19
2.2.5 Area of interest for the implantation 21
2.3 Data acquisition: recording techniques 21
2.3.1 Non-invasive techniques 22
2.3.2 Invasive techniques 22
2.3.3 Single-unit and multi-unit recordings 23
2.4 External devices and classification 25
2.4.1 Previous animal experiments 26
2.4.2 Primates 26
2.4.3 Rodents 29
2.4.4 Prediction algorithms 29
2.5 Feedback and plasticity 30
2.5.1 Motor imagery 30
2.5.2 Plasticity 30
2.6 The rat as a model 30
2.6.1 Cortical organization in the rat 31
2.6.2 The motor cortex of the rat 31
2.7 Summary of the problem analysis 33

Chapter 3 Problem Formulation 35
3.1 Problem formulation 35
3.2 Limitations 35
3.2.1 Development of a real time algorithm and external control device 35
3.3 Solution strategy 35

II Problem Solution **37**

Chapter 4 Experimental Protocol **39**
4.1 Behavioural training . . . 39
4.2 Materials . . . 40
4.3 Experimental setup . . . 40
4.4 Implant procedure . . . 41
4.5 Recordings . . . 42
4.6 Good channels detection . . . 42

Chapter 5 Data Analysis **45**
5.1 Data processing algorithm . . . 45
5.2 Time windows: Hit and NoHit data . . . 45
5.3 Denoising . . . 46
5.4 Spike detection . . . 47
5.5 Feature extraction . . . 48
5.6 Artificial neural network design . . . 49
5.6.1 Build the network . . . 50
5.6.2 Input vector and target vector . . . 50
5.6.3 Dividing the data . . . 53
5.6.4 Training the network . . . 54
5.6.5 Testing the network . . . 54
5.7 Summary of ANN design choices . . . 56
5.8 Support vector machine classification . . . 57
5.9 Summary of the SVM design choices . . . 58
5.10 Evaluating the classifier performance . . . 58
5.11 Misclassification error rate . . . 59

Chapter 6 Results **61**
6.1 Classify denoised and not denoised data with ANN . . . 61
6.2 Classify denoised and not denoised data with SVM . . . 63
6.3 Comparison between ANN and SVM classifiers . . . 65

Chapter 7 Discussion **67**
7.1 Methodological considerations . . . 67
7.2 Data Analysis . . . 68
7.3 Results . . . 69
7.4 BCI improvements . . . 69
7.5 Future prospectives . . . 70

Chapter 8 Conclusion **71**

Bibliography **73**

A Appendix **77**
A.1 The Cerebral Cortex . . . 77
A.1.1 Lobes . . . 77
A.1.2 Layers . . . 78

B Appendix **81**
B.1 Artificial neural network 81
B.2 The neuron 82
B.3 Models of the neuron 82
B.4 McCuoolgh and Pitts neural networks 84
B.5 Gradient Descendent Methods 85
B.6 Backpropagation 85
B.7 Different types of neural network 86
B.8 Initializing Weight 86
B.9 Feedforward network 86
B.10 Training the network 87

C Appendix **89**
C.1 Support vector machine 89

D Appendix **94**
D.1 Additional tables and results 94

Introduction 1

Traumatic lesions of the central nervous system (CNS), as well as neurodegenerative disorders, such as amyotrophic lateral sclerosis (ALS), brain stem stroke, muscular dystrophy, cerebral palsy (CP) or "locked-in" syndrome are causes of severe motor deficits in a large number of patients. Every year, spinal cord injuries are responsible for the occurrence of 11,000 new cases of paralysis in the United States alone. These cases have to be summed to the over 200,000 estimated population of patients who have to cope with partial (paraplegic) or almost total (quadriplegic) permanent body paralysis only in the United States. [Nobunaga et al. 1999][Carmena et al. 2003]

Considerable therapeutic interest is given to the option of restoring the voluntary motor control in patients suffering from traumatic or degenerative lesions of the motor system. In fact, quadriplegic patients suffer from sever damage of the central nervous system that heavily limits their every-day life. These patients cannot move any of their limbs and muscles below the neck. For this reason any help, or form of communication with the external environment, can provide a big increment of their quality of life. Until very recently, most of the focus from the research on restoration of motor functions was directed to repair the damaged axons that mediate the communication and therefore the motor neurons or alpha motor neurons in the gray matter of spinal cord [Ramón-Cueto et al. 1998]. Despite great effort it is still not possible to regenerate a large number of neurons to the original connection.

Two decades ago, an alternative method of rehabilitation for severely paralysed patients was introduced by Edward Schmidt [Schmidt 1980]. This approach suggested that direct interfaces between subcortical motor centres and artificial actuators could by-pass the spinal cord injuries allowing the patients to enact voluntary intentions. The challenge introduced by Schimdt assumed that voluntary commands can be translated in real time from motor cortex to directly stimulate the musculature of the patients or an external prosthesis [Carmena et al. 2003]. In fact, although they cannot move, they can still think about movement. Recently there has been considerable progress in designing prosthesis to assist this sort of patients. The goal of designing a brain computer interface (BCI) system is therefore to record these movement intentions, interpret them and use them to control an external device [Andersen et al. 2004].

Therefore, the main purpose of a brain computer interface (BCI), also called brain machine interface (BMI), is to interface the brain with an external device, such as a prosthesis (e.g. robotic arm), a wheelchair or a screen, to give to impair, partially paralysed or completely paralysed

patients the chance to communicate or interact with the world [Vallabhaneni et al. 2005]. The operative strategy of such an interface is to use the firing rates of cortical neurons and correlate them to continuous movement parameters or specific spatio-temporal spikes patterns [Olson et al. 2005]. A BCI creates in this way a direct communication pathway between the brain and an actuator independently from the normal output way based on peripheral nerves and muscles system to help paralysed patients to communicate with the surrounding environment.

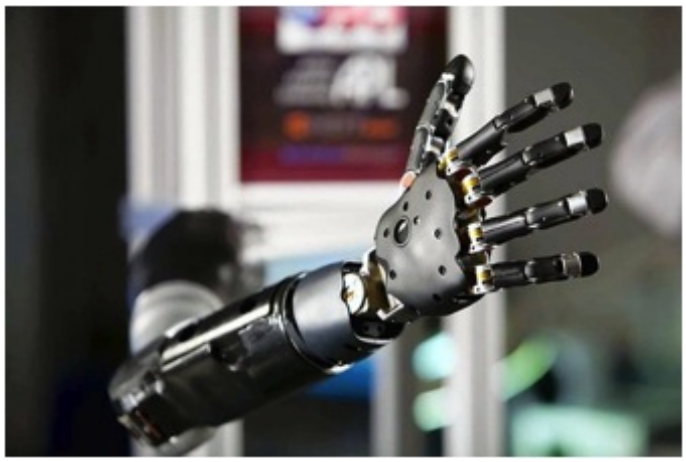

Figure 1.1. The DARPA Arm Johns Hopkins University. Example of prosthetic device electronically commanded by the intentions of the user.

1.1 Initiating problem formulation

To analyse the components of a BCI system in order to design a BCI system based on intra-cortical recordings.

Part I

Problem Analysis

Elements of a BCI 2

In order to implement a brain computer interface (BCI) system it is important to understand the principal components which form the BCI itself. Different types of BCI can be developed according to different types of rehabilitations and aims. A problem in the past was represented by the speed of the interface in processing the signals. A series of experimental and technological breakouts led to a new electrophysiological methodology for chronic, multi-site, multi-electrode recordings. After these improvements several studies have demonstrated that neuronal decoding of tactile stimuli can be performed real time by using pattern recognition algorithms, such as artificial neural network [Ghazanfar et al. 2000]. BCIs may lead to the definition of new experimental model, the real time neurophysiology. In fact, a BCI acts to investigate the real time operation of neural circuits in behaving animals. Grate efforts was spent in the last 10 years in recording, recognizing, differentiating and classifying different "intentions" in performing the movement. In particular, signals from the motor cortex were used to mimic a body movement, necessary not only to achieve a certain action but also in establish a good communication in patients suffering from a "locked-in" syndrome.

2.1 Principal components of a BCI

In a real world application a generic BCI system is formed by a person (the user) controlling a device (e.g. wheelchair) in an operating environment through a series of functional components. A set of functional components between the user and the device is considered as the BCI interface technology. The BCI interface technology is developed to help a target population with specific ability to perform certain tasks with a device [Mason & Birch 2003]. A general Brain Computer Interface system (BCI) is composed by four major parts:

1. The **data acquisition system**, which records the neural activity from the brain. It consist in an interface (e.g. electrodes) that can be implanted in the cortex or externally in the scalp and the device that effectively records and stores the signals. The recordings can be invasive or non-invasive; the signal can be of different types (e.g activity recorded from single or multiple neurons, analogue neural population signal like field potentials ecc.). Depending on the nature of these recordings the BCI can be implemented for different applications.
2. A **signal processing algorithm** which analyses and interprets the neural signals as control commands. The signal processing part links the recordings to an effector. It

determines which features from the recorded neural activity will be employed and which control commands should be created by the features.

3. The **external device** which is used as effector and controlled directly by the neural signals. It can be a visual signal (e.g computer cursor) or a complicated robotic or prosthetic system.
4. A **feedback** signal (e.g. visual or audio) proportional to the neural activity sent from the device to the user in order to improve the brain plasticity and the accuracy of the movement.

[Waldert et al. 2009]

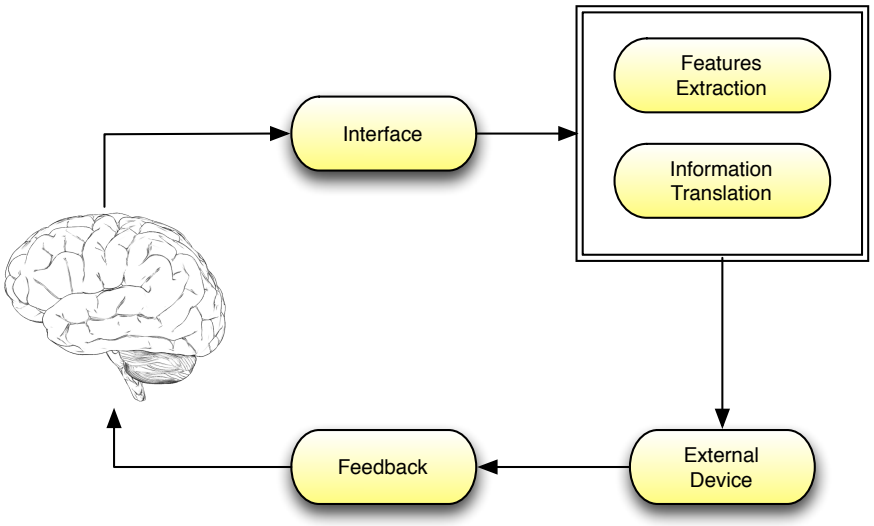

Figure 2.1. Functional components of a BCI system: an interface to record the brain activity, a mathematical model, or algorithm, to analyse and interpret the signal, an external device used as effector, that can be a cursor in a screen or a robotic arm, and a feedback signal.

The following sections contain the a detailed description of the principal components of a BCI system. This background was important to have a general overview about different solutions for the implementation of the BCI, to better understand the choices to be made in the implementative part. A block diagram of a BCI system is presented in Figure 2.1. It represents the principal components of a BCI, starting from the brain, where the signal is collected by an interface. Afterwords the signal is processed, in particular the features are extracted and translated into information and control commands. The control commands are sent as input for an external device, that gives to the user a feedback, useful to increment the plasticity and to learn how to command the effector. Section 2.2 gives a brief description of the parts of the brain involved in movement control and the main areas of interest for extracting motor signals. Furthermore how the voluntary movement is encoded by cortical neurons is described, to understand the types of interfaces (e.g. electrodes) and their locations. Section 2.3 presents the data acquisition strategy, an overview about the possible signals generally and historically used to decode movement parameters, the advantages and disadvantages of their employment, and their information content. Sections 2.4 provides a description of the signal processing part and the classification methods, through a brief review of previous studies, giving an overview about

possible effectors (e.g. robotic arm and computer cursor). Section 2.5 describes the motor imagery and the feedbacks available to improve the brain plasticity. In the end, in section 2.6 the laboratory rat is presented and some considerations about the reliability of the chosen model are made.

2.2 The motor cortex

In decoding motor signals from the brain, it is important to understand how the voluntary movements are encoded by the cortex, and where they are generated. The voluntary movements are organized in the motor cortex. In particular the primary motor cortex is the final output stage following a processing taking place in many other brain areas. The discovery, in the early 1870s, that electrical stimulation of the frontal lobe in different species results in movements of the controlateral side of the body, had a great impact in neurological studies. The resulting motor map was correlated with previous clinical and anatomical observation of effects of local brain lesions. In this way it was discovered that the precentral gyrus, already Broadman's area 4, what we call now primary motor cortex (M1) is the area that needs less electrical stimulation to react with a correspondent movement. The motor map showed an orderly arrangement of human body, called *homunculus* (from latin "small man"), along the *gyrus* corresponding to face, digit, hand, arms, trunk, leg and foot. Fingers, hands and face, that are the parts of the body which require grate movement precision, correspond to a bigger surface in the homunculus, that is not proportional to the human body representation.

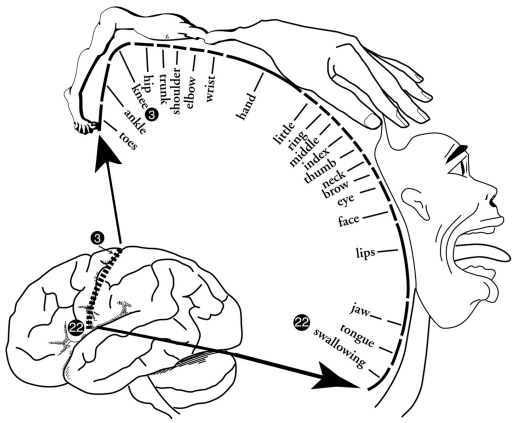

Figure 2.2. The homunculus shows the body representations in the primary motor cortex.

The early experiments of electrical stimulation led to the idea that the primary motor cortex was an exactly correspondent representation of the body, and that there was a correspondence between certain groups of neurons and muscles. However, after more accurate study, they discover that it was not true. In fact the same muscle can be controlled by different areas in the motor cortex, as the same area is concentrically organized and can potentially control more than one muscle. [Kandel et al. 2000]

In general, the motor cortex is divided into different regions characterized by different functional organizations and reactions to electrical stimulation.

2.2.1 The primary motor cortex

In the posterior portion of the frontal lobes along the central sulcus is located the primary motor cortex, as shown in Figure 2.3, that is important in voluntary movement. This area works in association with the premotor cortex and the supplementary motor area. The primary motor cortex projects through axons from neurons in layer V into the spinal cord to synapse into the interneuron circuitry and directly onto the alpha motor neurons in the spinal cord, connected to the muscles. It contains a partially overlapped map of the body.

The primary motor cortex has two levels of organization, a *low level*, where the corticomotor (CM) cells control groups of muscles corresponding to a precise control of action, and a *higher level* encoding system that corresponds to more global features of movement. An example of this can be found in the movement of the digits. Although individual neurons fire maximally with a movement of single digits, the digits control is spread along the cortex. In fact current evidence demonstrate that CM cells control small groups of muscles, and the ones involved in individuated finger movements have axons that diverge to more than one motor nucleus in the spinal cord. This is due to the fact that a contraction or movement in a single digit influences also all the other. In particular was found that for two different task, grip force and precision, involving the same muscles different patterns of firing are required. And usually certain cortical cells fire less and less often as muscle force increase. [Kandel et al. 2000]

2.2.2 The premotor cortex

In the late 1930s was discovered that not only a stimulation of the Broadman's area 4, but also the stimulation of Broadman's area 6 provoke a motor reaction, though with a grater stimulation intensity. This area is situated anterior to the motor cortex, and has the same pyramidal neurons in layer V. These neurons project directly to M1 and the spinal cord, but with smaller dimensions and with less connections. The stimulation in this area, unlike the motor cortex, provokes a more complex movement involving multiple joints and resembling coordinated movements. Moreover a stimulation in this area results sometimes in a bilateral reaction, suggesting that this area is involved in coordination of the movement on the two sides. Evidence that premotor area and primary motor cortex are different although their domains overlap in the spinal cord are that their inputs are quite different. The premotor area area receives input from the supplementary motor area, that seems to be involved in learning the sequence by memory and it fires only once the movement is learned. The premotor area seems to work in a more complex way, a damage in this region will provoke a complex motor impairment, related to incorrect visuospatial processing and coordination. Moreover, many studies in humans and monkey demonstrate that neuronal population from the premotor cortex also connected with ipsilateral, bilateral and combination of movements. Movements initiated by external events involve the lateral premotor area, as task like delay action or mapping relationship between stimulus and response. [Kandel et al. 2000]

2.2.3 The supplementary motor area

The supplementary and pre-supplementary motor area are located on the top of the dorsal part of the cortex, as described by Penfield & Welch [1951]. Each neuron in this area projects in many muscles in both sides of the body. For this reason the map of the body in the supplementary motor area overlaps. This part of the cortex is thought to play a role in planning movements, and in particular for internally generated plans involving sequence of movements. This was proved by the observation of a preparatory potential, a negative potential seen in the electroencephalogram (EEG), recorded from this cerebral region. This negative potential has a characteristic negative shift appearing nearly 1 s before the movement, demonstrating that planning occurs before movement. Neurons in this region usually fire only for performing task already memorized. On the contrary neurons from the primary motor cortex will fire with the same degree before and during guided and memorized tasks. Thus the supplementary motor area (SMA) seems to be involved in programming movements from memory. Other evidences about the role of the (SMA) come from experiment of imaging technique in human brain, where Roland suggested that SMA was active during internally generated plans and experiment in monkeys [Roland et al. 1980]. These experiments quantified the neuronal activity using the blood flow measured by estimating the radioactivity. Raster plots of activity in the motor cortex showed that cells in the SMA fire only when a trained sequence was performed. [Kandel et al. 2000, chap. 38]

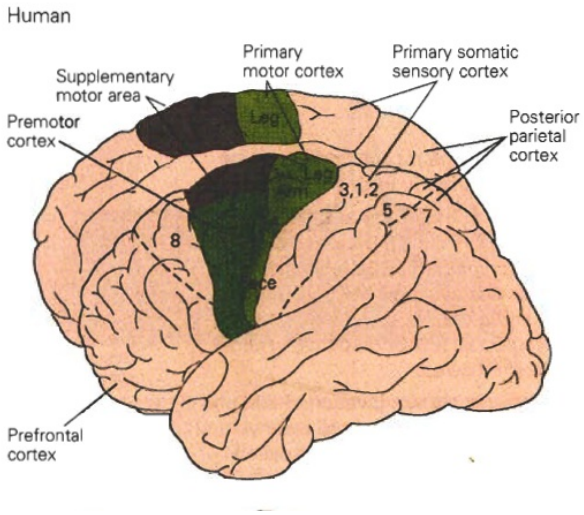

Figure 2.3. Motor cortex division in human brain: primary motor cortex, premotor cortex and supplementary motor area [Kandel et al. 2000].

2.2.4 Role of the motor cortex in voluntary movement

The motor system is organized in a functional hierarchy, with different levels representing different decisions. The most abstract level concerns in the *purpose of the movement*, and is represented by the dorsolateral frontal cortex. The next level concerns with the *formation of a motor plan*, and is obtain from an interaction between posterior parietal and premotor areas. The premotor cortex specifies the spatial characteristics of a movement based on the information coming from the sensory area about the environment and position of the body in the space. In a lower level appends the *coordination of spatio-temporal details* of muscles

contraction, needed to execute the planned movements. The coordination is executed by the motor circuits in the spinal cord. A single movement is the sum of different parallel pathways that project to the spinal cord. Above the spinal cord there is the brain stem and above the brain stem there are the cerebellum and the basal ganglia, for modulating the actions. [Kandel et al. 2000]

2.2.4.1 Encoding movement

The principal question to answer is whether the neurons encode specific spatio-temporal information, or more general features of the movement like direction, extent and angle changes. Another question is how can neurons that are so broadly tuned encode a precise information. Two different approaches can be followed to answer these questions, on the one hand the theory of Evarts, introduced almost forty years ago, that correlated the neuronal activity with some variables such as movements and force at individual joints [Evarts et al. 1984]. On the other hand a second approach, introduced by Apostolos Georgopoulos, based on the movement reaching for a target, rather that based on individual joints. [Scott 2000]

Georgopoulos et al. [1986] tried to investigate in monkeys the brain mechanisms subserving the direction of the hand movement in a three dimensional (3D) space in order to predict it. They studied trained monkeys moving a joystick in different directions. It was found that the movement in a certain direction was determined by the contribution of a large population of neurons, considered as a relevant unit in the brain, rather than the single neuron. The direction was found to be uniquely predicted by action potentials of neurons in the motor cortex. Although neurons in the arm area of the primate motor cortex are only broadly tuned in this direction, the animal can control the movement precisely.

He first purposed a vectorial representation of the neural population, Figure 2.4, in which each neuron corresponded to a vector, with a length proportional to the single neuron activity, that made a weighted contribution along the axis of its preferred direction. The frequency of discharge was used as a measure for the neural activity. The cell preferred direction was the direction along which a neuron seemed to react better, and had a greater activity. The broad directional tuning of a single neuron suggested that the direction of the movement was not coded by individual cell that responded only to specific directions of movement, but was the result of the contribution coming from the entire population. This theory was supported by the result that the population vector direction was very close to the actual direction of the movement. [Kandel et al. 2000]

Subsequent works seemed to demonstrate impressive relationship between hand motion and neural activity, at both single cell and population levels. These correlation was interpreted by thinking the motor cortex as a higher level of features integration, rather than a lower level, useful for joints and muscles movements. This interpretation suggested a hierarchical organization of the motor system, above (premotor cortex) and below (spinal cord) the motor cortex. For example the primary motor cortex (M1) represents the hand movement with a population code, transformed into command to the specific muscles in the spinal cord level [Scott 2000]. Other studies investigated the level of movement control exerted by the primary motor cortex (M1) proved its crucial role in the production of all voluntary movements by demonstrating the almost complete paralysis followed by M1 lesions.

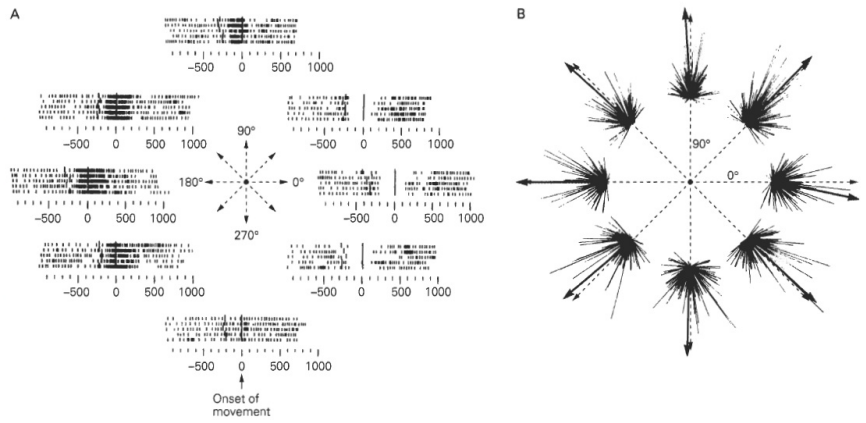

Figure 2.4. The figure shows the experimental results achieved by Georgopoulos et al. in 1982 [Georgopoulos et al. 1986]. The direction of movement was encoded in the motor cortex by the pattern of activity of the population cells. Motor cortical neurons were broadly tuned but individual neurons fired accordingly to a preferred direction. A) the figure presents raster plots of firing pattern of a single neuron during movement in 8 directions. B) The figure shows that cortical neurons with different preferred direction were all active during movement in a particular direction.

A study written by Todorov [Todorov et al. 2000] reported that in awake behaving monkeys the activity of most M1 pyramidal tract neurons was directly related to the amount of force exerted. This led to the idea that the same cells that encode hand velocity movement could also encode the force exerted against an external object. In other studies M1 was also correlated with hand position, acceleration, movement preparation, target position, distance to target, overall trajectory, muscle coactivation, serial order, visual target position and joint configuration.

2.2.5 Area of interest for the implantation

The quantity and the location of neural tissue to interface are directly linked with the characteristic of the motor commands that have to be extracted. Although it is well known that cortical neural activity can encode different motor parameters, it is not clear which cortical area can provide the best inputs for a BCI. Due to the distributed nature of the motor planning, it is also not clear if the primary motor cortex (M1) [Chapin et al. 1999] or the parietal cortex [Pesaran et al. 2000] should be used for the main input. Although cortical areas are know to have cortical specializations was observed that all single neurons located in frontal and parietal areas contributed, at different levels, to the prediction of all parameters analysed [Carmena et al. 2003].

2.3 Data acquisition: recording techniques

The electrophysiological signal to be processed can be of various nature. For example, it can involve recordings of action potential of single neurons or recordings of neural activity of a large groups of neurons. Either non invasive or invasive techniques can be used, Figure 2.5.

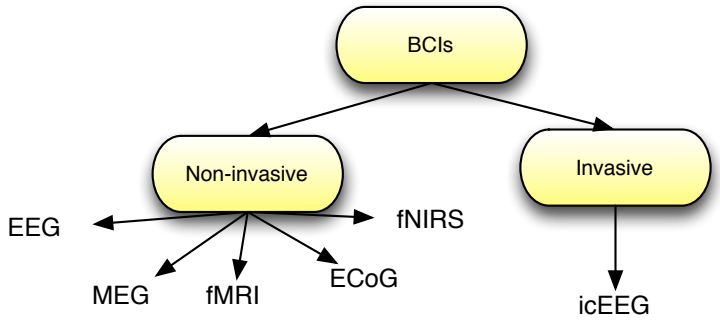

Figure 2.5. Block diagram summarizing the two different types of recording techniques: non-invasive and invasive.

2.3.1 Non-invasive techniques

Non-invasive recordings consist in measuring the signal without penetrating the tissue. For example electroencephalograms (EEG) signal recorded on the scalp, which reflects extracellular currents, or magnetoencephalograms (MEG) signals, which reflects intracellular currents flowing throw dendrites. This approach is proved to be effective in helping "locked-in" patients giving an alternative communication channel. Both of them reflect the activity of a large population of neurons. Alternatively, functional magnetic resonance imaging (fMRI) and functional near-infrared spectroscopy (fNIRS) measure neural activity indirectly, based on the blood oxygenation level, showing a low temporal resolution. Despite having the advantage of giving no risk and avoiding the surgical parts, they have a limited spatial resolution (EEG) and temporal resolution (fMRI) in recording signals. [Waldert et al. 2009]

EEG

Generally, EEG-based BCI try to predict the subject's intentions and decisions through a combined measure of electrical activity from a massive neural population recorded along the scalp. This technique presents a low spatial and temporal resolution, due to an overlap of information coming from multiple cortical areas and also due to the low pass filter effect, exerted by the tissue, bones and skin. EEG is also affected by mechanical artifacts, which come from the movement of muscles and eyes. Despite these well known limitations, EEG technique can detects modulation of brain activity, correlated with cognitive states, voluntary intentions and visual stimuli. [Lebedev & Nicolelis 2006]

2.3.2 Invasive techniques

The most spatially resolved information is achieved from implants going into the cortex. Invasive techniques consist in interfacing directly with the cerebral cortex. In fact, the invasive electrodes are implanted intracranically and provide signals of the best quality, representing a big potential for further improvements. They can record single cells brain activity (SUA) or the activity of multiple neurons (MUA). A third type of invasive technique is the local field potential (LFP). SUA is obtained by high-pass filtering (>300 Hz) the extracellular potential, spike detection and spike sorting, with the aim to assign each spike waveform to a corresponding neurons. MUA is obtained in the same way without spike sorting, therefore the signal consists

in activity originating from multiple neurons. LFP is extracted by low-pass filtering (<300 Hz) of the extracellular potentials, and reflects the synaptic input of the neural population in the proximity of the electrode tips. [Waldert et al. 2009]

LFP

Neurons produce action potentials referred as *spikes* in laboratory jargon. The electrode tips record components due to the synaptic currents and action potentials. The synaptic currents have slower time course,on the other hand the spikes have faster time course. For this reasons they can be separated by an high-pass filter (for the spikes) and low-pass filter (for the synaptic mechanism). The component due to the synaptic mechanism is what is known as the local field potential (LFP). The LFP is therefore an electrophysiological signal obtained by recording the summation of neural activity. It is considered to arise from the excitatory and inhibitory dendritic potentials and thus serves as a marker of inputs and local processing of a wide volume of brain tissue extending several hundred micrometers.

ECoG

The electrocorticogram (ECoG) is recorded by subdural electrodes, with invasive technique, or also epidurally, with non invasive technique. In the case of recordings with invasive technique, it reflects synaptic inputs to a neural population typically located within 100 μm beneath each electrode tip [Waldert et al. 2009]. It samples a smaller cortical area compared to EEG, resulting in more accuracy and shorter training times. It is, like LFP, an analog signal.

The different types of signal, which can be recorded from the brain are summarized in Figure 2.6. In the figure is specified whether the technique used is invasive or not, the spatial resolution and the size of neural cluster.

2.3.3 Single-unit and multi-unit recordings

Nowadays it is accepted that highly distributed populations of broadly tuned neurons can sustain a continuous production of motor behaviours in real-time. The current idea of sampling the extracellular activity from a large population of individual neurons emerged in the 1980s, replacing the conviction that single-neuron was the functional key of neural information. The idea of the *population coding* firstly proposed by Young [1802] and Hebb [1949], is now supported by the evidence that distributed ensembles of neurons define the true physiological unit for the mammalian CNS. In particular increasing the size of neural population will increase the quality of prediction. The importance of single neuron activity is still demonstrated by the fact that it can be conditioned to produce firing patterns whether neuronal activity is presented to primates as sensory feedback. In some experiments the firing of a single cell becomes well correlated with the desired pattern, that can be used to control movements. [Nicolelis & Lebedev 2009]

Therefore BCI provides a new insights into important questions dealing with the issue of how information is processed by the central nervous system (CNS) during the generation of motor behaviours. BCI studies allowed Nicolelis & Lebedev [2009] to formulate a series of principles

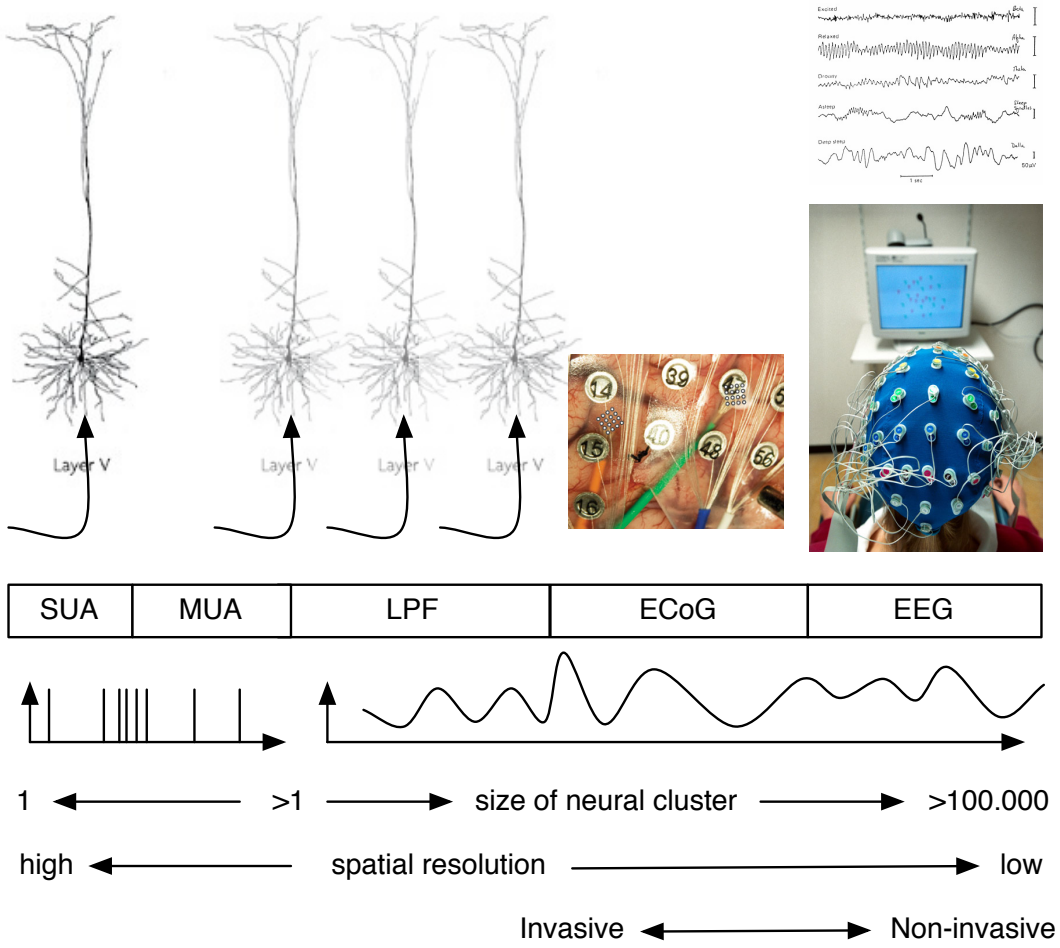

Figure 2.6. The figure shows different types of non-invasive and invasive recording techniques of neural activity. In the bottom part, are specified whether the signal is digital or analogue, the size of the neural cluster and the spatial resolution. [Waldert et al. 2009]

that may also be used to implement a BCI and therefore a neuroprosthetic device. One of the interesting question to answer about the implant, and indirectly about the nervous tissue is whether any changes, like a small reduction of the original population, natural loss or death of the recording cells, could impede the reliable function of the implant. Furthermore, the effectiveness of the implant can be also reduced by the change in physiological properties. One possible limits is to maintain the daily performances for many years.

BCI principles

To overcome these problems and in order to better understand the neural population coding, Nicolelis & Lebedev [2009] proposed eight principles regarding BCI:

I *The distributed coding principle*
The representation of any behavioural parameter is distributed across many brain areas

II *The single neuron insufficiency principle*

Single neurons carry only a limited amount of information about a given motor parameter

III *The neuronal multitasking principle*
A single neuron is informative of several behavioural parameters

IV *The neuronal mass principle*
A certain number of neurons in a population is required for their information capacity to stabilize at a sufficient high value

V *The neural degeneracy principle*
The same behaviour can be produced by different neuronal ensembles

VI *The plasticity principle*
Neuronal ensemble stays constant during the learning of a task. Neural ensemble function is crucially dependent on the capacity to plastically adapt to new behavioural tasks

VII *The conservation of firing principle*
The overall firing rates of an ensemble stay constant during the learning of a task

VIII *The context principle*
The sensory response of neural ensembles changes according to the context of the stimulus

Usually, multiple computational models are employed to extract motor parameters (e.g. position, velocity and gripping force). Computational models are first trained to predict motor parameters from neural ensemble activity while the subject is performing a motor task. Then a "transform function" is derived for motor pattern belonging to a particular movement. The next step allows directly the brain (brain control) to control the movement of an external device. [Nicolelis & Lebedev 2009]

To provide a control signal, motor parameters (e.g. hand trajectory) and cognitive parameters (e.g. the goal and the predictive value of an action) can be decoded from the brain activity. A neural prosthesis using both cognitive and motor parameters can ideally achieve a maximized level of communication with the outside world [Andersen et al. 2004]. Therefore the two main types of available prosthesis, which capture more scientific attention, are the motor prosthesis and the prosthesis for communication.

2.4 External devices and classification

Usually, multiple computational models are employed to extract motor parameters (e.g. position, velocity and gripping force). Computational models are first trained to predict motor parameters from neural ensemble activity while the subject is performing a motor task. Then a "transform function" is derived for motor pattern belonging to a particular movement. The next step allows directly the brain (brain control) to control the movement of an external device. [Nicolelis & Lebedev 2009]

To provide a control signal, motor parameters (e.g. hand trajectory) and cognitive parameters (e.g. the goal and the predictive value of an action) can be decoded from the brain activity. A neural prosthesis using both cognitive and motor parameters can ideally achieve a maximized level of communication with the outside world [Andersen et al. 2004]. The motor prostheses aim is to guide paralysed or prosthetic arm in order to reach a certain movement. On the contrary, the aim of a communication prosthesis is to process brain signals in order to guide a prosthesis to arrive at the end point following a certain trajectory, path or speed characterizing the movement. Therefore, depending on the BCI system aim, different types of classification and external devices can be used.

2.4.1 Previous animal experiments

Initial experiments about BCIs came from studies conducted by Fetz [1969] demonstrating that macaque monkeys could learn to selectively adjust the firing rate of its cortical neurons. Animal experiments were conducted in primates by Wessberg et al. [2000] Serruya et al. [2002]Taylor et al. [2002], rodents [Chapin et al. 1999] and also in human subjects [Birbaumer 2006]. One dimensional cursor movement was accomplished using spikes activity from a paralysed human subject by Kennedy & Bakay [1998]. Some successful examples of neuro-engineering implementing a simple BCI are brain stimulators such as the cochlear implant [Wazen et al. 2003], deep brain stimulators for Parkinson's disease [Limousin & Martinez-Torres 2008] and vagal nerve stimulators for treating epilepsy.

Recent studies demonstrated that monkeys can control the displacement of a cursor in a computer and accomplish 1D or 3D movements of a simple and elaborate robot arm without the animals making any movement. In particular signals related to the grip force, were extracted and decoded to control the size of the cursor [Carmena et al. 2003]. Olson et al. [2005] demonstrated, studying rodents, the possibility of deriving information on a paddle pressing task, mimicking the control of a wheelchair turning left or right, by cortical recordings in Sprague-Dawley rats.

2.4.2 Primates

Real-time prediction of the hand trajectory

Wessberg et al. [2000] tried to predict a real-time hand trajectory from the cortical neurons in primates. Microwires were implanted in multiple cortical areas (premotor area, primary motor area, posterior parietal cortical area) of two owl monkeys to record cortical neurons ensembles. The monkeys were trained to perform two different tasks: 1D hand movement to displace a manipulator in two different directions following a visual cue and a 3D hand movement to reach a piece of food randomly placed in four position on a tray. The experimental design is presented in Figure 2.7.

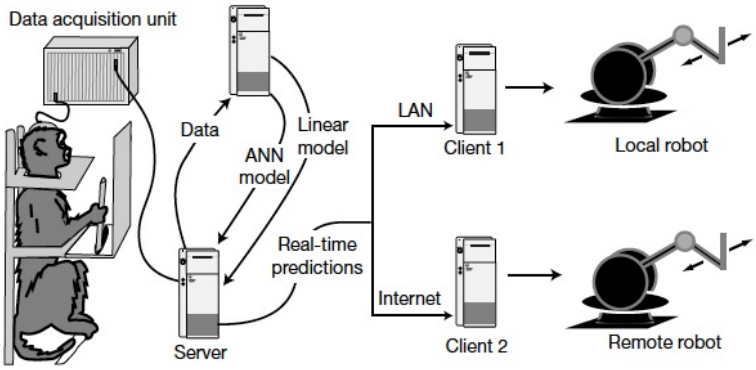

Figure 2.7. Schematic diagram depicts the apparatus employed by Wessberg et al. [2000] to record cortical ensemble data from primates. Both linear and ANN models are continuously updated during the recording session.

Predictions of the hand trajectories (Figure 2.8) were based on signal generated by a population of single neurons. A correlation analysis demonstrated that the activity of most single neurons correlate with both 1D and 3D hand trajectories. Although the degree of frequency of discharge was different within and between cortical areas. Both linear and non-linear (ANN) models were applied to the data. Several feedforward ANNs were evaluated off-line. The best results were obtained using 15-20 neurons in the hidden layer, linear output units, the Powell-Beal conjugate gradient training algorithm and early stopping rules. During real time prediction of the hand position, up to 10 min of data were first collected to fit the linear model and the ANN model. To reduce the influences of dynamics changing in coupling between movement and neural activity the models were continuously adapted with the last 10 min of recordings. Each neuron discharge was counted in 100 ms bins. No large differences in fitting accuracy were observed between linear and ANN algorithms.

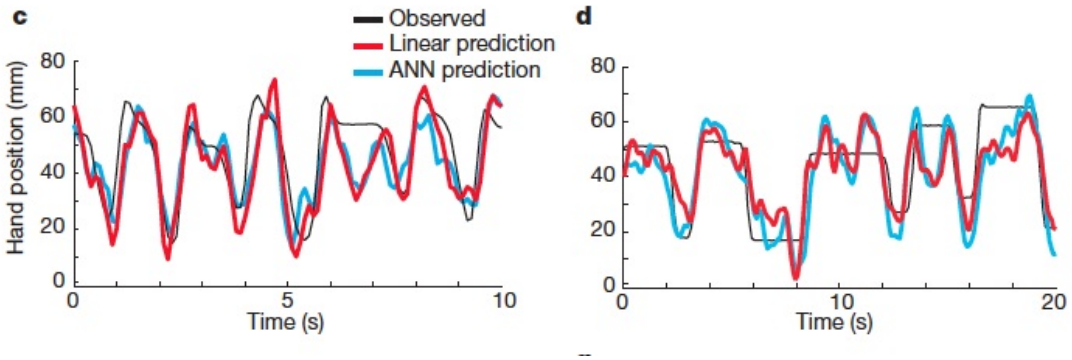

Figure 2.8. Observed (black) and real time predicted 1D hand movements using linear (red) and ANN (blue) models. [Wessberg et al. 2000]

Real-time prediction of grip force

Carmena et al. [2003] tried to predict the hand trajectory and the grip force using a similar behavioural design as Wessberg and colleagues, see Figure 2.9. The monkeys were trained to

move the arm and to grip a pole to move a cursor in a screen and to change its size, and later to perform the two tasks together. The experiment was performed following two modalities, the "pole control" and the "brain control" mode. In the first case the monkeys performed the actual movement, in the second the movement was only thought. Multiple arrays containing 16-64 microwires each were implanted in several frontal and parietal cortical areas. The neurons activity was simultaneously recorded from single neurons and multi-unit during each recording session.

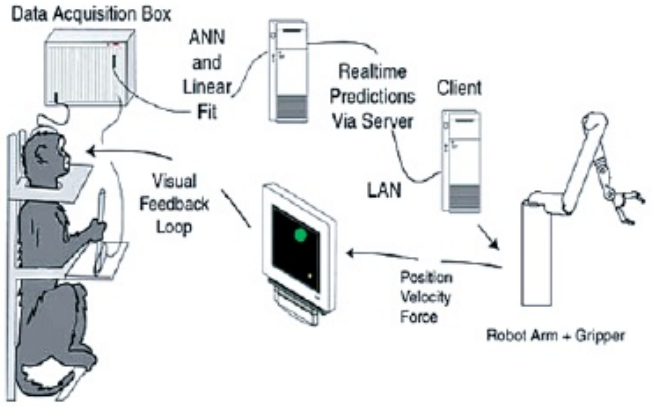

Figure 2.9. The figure shows the behavioural setup and control loops, consisting of a data acquisition system, a computer running liner models in parallel and real time, a robotic arm and a screen providing as a visual feedback.[Carmena et al. 2003]

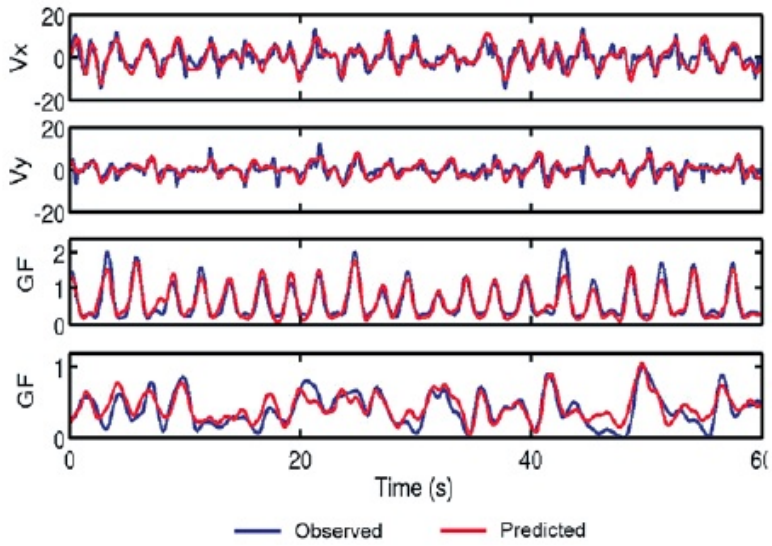

Figure 2.10. Comparison of recorded motor parameter (blue) with parameter predicted by the linear model (red). The velocity is reported in its components as Vx and Vy, and the grip force as GP during the execution of task 1 and 2. [Carmena et al. 2003]

Multiple linear models were used to simultaneously extract a variety of motor parameters (e.g hand position, velocity and gripping force). The gripping force was modelled as a weighted linear combination of neuronal activity using a multidimensional linear regression or Wiener

filter. Neuronal firing rates were sampled using 100 ms bins, and 10 bins preceding a given point were used to train the model and predict with it. Models were trained with 10 min of data and tested by applying them to subsequent recordings. Several alternative decoding algorithms were tested off-line, including normalized least mean square filter, Kalman filter, feedforward backpropagation ANN. However, better results were not achieved by that. Grasping force feedbacks were provided. A 6 degree of freedom (DOF) and 1 DOF gripper were used.

2.4.3 Rodents

Olson et al. [2005] proposed an alternative system where specific spatio-temporal spikes pattern were used to detect classes of behaviour with the aid of non linear classification algorithms, such as support vector machine (SVM). The motor cortical activity from the rats were not used to update a second-by-second 3D trajectory but was applied to higher level commands. This technique can be useful for applications requiring steering a wheelchair "left" or "right" and letting sensors and actuators to work on the trajectory itself. This can be seen as a series of simple asynchronous decisions driving towards a goal. The driving surrogate task consisted in a binary paddle pressing movement in a conditioning box mimic the trajectory depending on different hitting directions (e.g. paddle on the left or paddle on the right). In fact little differences were noticed when the animal pressed two identical paddle varying only by their spatial location. A 2 by 4 array was chronically implanted in the rat motor region, independently from the preferred hand. Action potentials were amplified and pre-processed. After the spike sorting procedure temporal bins were extracted for each neuron and summed into time windows, containing the number of events taking place in every bin. Support vector machine classification was performed. To obtain a quantitative evaluation a leave-one-out cross validation was implemented.

2.4.4 Prediction algorithms

Prediction algorithms are the heart of all closed loop BCI studies. Two broad classes of prediction algorithms typically used are *regression*, to predict a continuous variable, and *classification*, to predict discrete classes. The output from the signal processing algorithm can be used afterwards as input for a classifier that determines different types of movement from a range of data. Therefore, a classifier makes an association between a range of data and two or more classes or conditions. It is defined as an input/output device where the input is a set of measurements, and the output is the most likely class associated with the data set. If the decision boundary, separating two classes, is a straight line, where two different spaces can be distinguish it is called linearly *separable*. If the data can be separated by a curvy line is called *non-linearly separable*. As previously seen in most of the experiments, the firing rate was considered the main discriminant feature to classify the movement and to provide a control command. For example, the monkeys reaching tasks can map neural signals into 3D velocity space, or predict the grasping force intensity. The neural activity of single neurons belonging to a population was extracted, after spike sorting, using firing rate features within 100 ms windows [Carmena et al. 2003].

2.5 Feedback and plasticity

In the past decades much efforts to develop BCIs were given to three main components: (1) the multi-electrode recording sensor array, (2) the decoding algorithm and (3) the output interface to be controlled by the cortical-derived signals. Much less attention was paid to the fourth component: the sensory feedback. A fundamental part of the BCI is the feedback mechanism, that allows to increase the plasticity and therefore to increase the performances of the implant. In a closed loop BCI a possible feedback can be a food/water reward [Olson et al. 2005], a visual feedback like a cursor in a screen [Carmena et al. 2003], a sensory feedback, like a vibromechanical stimulation, or even an auditory feedback. Suminski et al. [2009] hypothesized that relevant proprioceptive sensory information are presented in the activity of neurons in M1, and that the addiction of multisensory feedback information about an observed action enhanced the congruence and modulation in neural activity. In fact, the activity in M1 is not only related to motor output but also to many types of movement information including spatial goals, hand motion, sensory feedback, force output and motor activity. [Suminski et al. 2009]

2.5.1 Motor imagery

The only way for a patient suffering from cerebral, spinal or peripheral damages of the nervous system to control a BCI is represented by the motor imagery. Motor imagery is the result of a conscious access to the content of intentions to move, which is usually an unconscious process part of the movement preparation. For this reason mental practice with motor imagery provides a performance improvement. Although evidences demonstrate that "first-person" kinetic motor imagery elicits a stronger activation of the motor cortex M1 compared with "third-person" visual motor imagery, the latter is the only one that can be integrated with a consistent feedback (visual and not proprioceptive) [Lotze & Halsband 2006]. A visual feedback can be used to build a map between neural modulation and cursor motion which is able to guide the movement of a BCI in patients who are not able to generate movements.

2.5.2 Plasticity

Experience dependent plasticity in cortical neurons is essential to learn to operate a BCI. The plasticity is characterized by the changes in tuning properties of individual neurons and physiological adaptation for the neural ensembles. Such changes refer to the connection strength and the gene expression. When plasticity occurs significant portion of recorded neurons progressively acquired properties related to the kinematic properties of the robotic device used. [Nicolelis & Lebedev 2009]

2.6 The rat as a model

The impossibility, for ethical reasons, to study the cortical morphology and functions in human subjects directly, and the need to investigate human cortical functions, led the man to conduct experiments in animals. Flourens, already in the 1823, was one of the first scientists who conducted various animal experiments with chicken, pigeon and dog to study the cortical functions after the ablation of various parts of the brain [LeDoux 2005].

Historically the methodological problems of how to generalize about human brain functions from the study of mammals is one of the most debated questions. The publication of *"Descendent of Man"* by Charles Darwin can be taken as the beginning of the widespread interest in non human subject as an alternative way to study physiological processes [LeDoux 2005]. Nowadays it is believed that most process in humans relay to that part of the brain called "neocortex" and that the cognitive complexity is directly linked with the cortical complexity.

Primates have played a central role in motor system and movement disorders research because of their very close neuroanatomical and neurophysiological similarity to humans. Neuroanatomical and biomechanical underpinnings of movements can also be investigated in rats, that seem to be very similar to primates as presented with legitimate arguments by Cenci and colleagues (2002) [LeDoux 2005, chap.A2].

The thesis supported by Kolb and Tees [LeDoux 2005] in their book *The Cerebral Cortex of the Rat* states that "rat provides a useful alternative to the primate and can serve as model for cortical function". Nevertheless it has to be considered that the choice of the animal is strictly dependent on the nature of the problems. Thanks to their dimensions rats have long been the animal of choice for studies of neurological disease (e.g. stroke, traumatic brain injury, epilepsy and multiple sclerosis) cerebral ischemia and cardiovascular physiology. In fact it is big enough to allow surgical procedure on major organs and chronic neural recordings with relative ease [LeDoux 2005, chap. A2].

The laboratory rat plays a predominant role in neuroscience research, and it is probably the most widely used subject. To better evaluate the choice of the rat it is important to compare the brain structure and physiology of this animal with human and to understand the limits of this comparison from the differences found.

2.6.1 Cortical organization in the rat

The central nervous system (CNS) of the rat is formed by the brain, consisting in cerebrum and cerebellum and the spinal cord. The CNS has three meninges like in humans: dura mater, arachnoid and pia mater. The brain and the spinal cord are bathered in cerebrospinal fluid. The mammalian neocortex is defined as the youngest part of the brain, developed last in the evolution. It is more or less uniform with a six layers structure. [Kolb & Tees 1990]

2.6.2 The motor cortex of the rat

In humans the motor cortex, can be divided into three main regions: the primary motor cortex (M1) and the secondary motor cortex, formed by the premotor area (PMA) and the supplementary motor area (SMA) [Kandel et al. 2000]. The motor cortex in the rat is located in the frontal neocortical region, formed by the agranular isocortical area Fr1 Fr2 and Fr3. This is proved by the agranular cytoarchitectonical structure of the frontal cortex, that suggests a motor function.

Whereas Fr1 and the rostral part of Fr2 contains many neurons in layer V projecting to the spinal cord, Fr3 and the caudal part of Fr2 contain only a few or none of these neurons. Fr3 can probably be identified as a motor projection of the jaw and the caudal part of Fr2 contains the

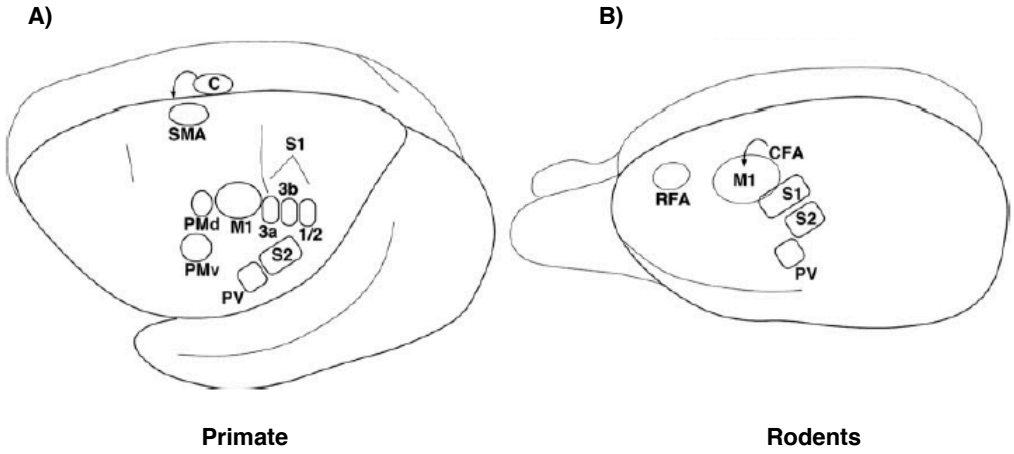

Figure 2.11. Comparison of the somatosensory and motor areas in primates and rodents. A) Represents the brain of the primate, with the somatosensory areas (S1, S2 and PV) and the motor area M1, PMd PMv, SMA and C. B) Represents the lateral view of the rodents brain with the corresponding S1, S2 and PV and motor area M1 CFA and RFA [Nudo 2007].

motor representation of the vibrasse and eye muscles [LeDoux 2005, chap. 4]. Different studies by Neafsey et al. [1986] report the same body area to be represented in different regions, suggesting a multiple representation to be an organizational principle of the rat cortex. This was proven by using intracortical stimulation mapping technique by which two forelimb motor areas where delineated, one corresponding to the traditional primary forelimb motor area, and another located rostrally that overlaps the separate rostal patch of corticospinal neurons near the frontal pole.[Neafsey et al. 1986]

Both primates and rodents (and presumably all mammals) have at least three somatosensory areas (S1, S2 and PV) and at least one motor area (M1). However, in primates the differentiation seems to be more complex for S1 and also M1, that posses additional motor areas, including dorsal and ventral premotor area (PMd) and (PMv), SMA and the cingulate motor area. Rodents posses the caudal forelimb area (CFA), corresponding with M1 in primates and the rostral forelimb area (RFA), similar to PMA and SMA, but with a different evolution in rats. [Nudo 2007]

Detailed mapping experiments after intracortical microstimulation of the rostral motor area revealed a small medialy located hind limb area, indicating a complete body representation may exist here, and suggesting that this rostral motor area may be the rat's homologue of the primate's supplementary motor area [Neafsey et al. 1986, chap.8]. One characteristic feature that distinguishes this functional map from the previous is the large amount of the sensory cortex that yields to a motor response at currents less than 50 μA. This does not mean that there is no distinction between the two, which differ from the sensory inputs they receive.

Therefore, the electrode was implanted to record the activity within the primary motor cortex (M1). The implantation site was estimated to be 2 mm behind the *bregma*, the conjunction point on the skull at which the coronal suture is intersected perpendicularly by the sagittal suture.

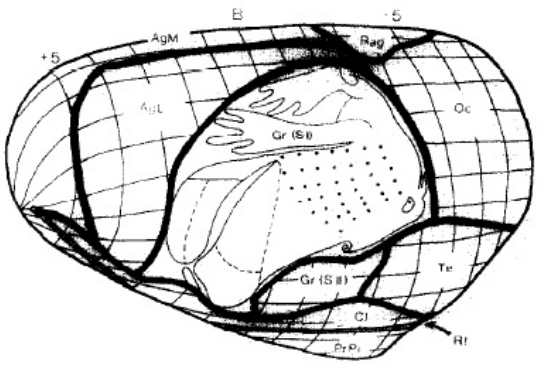

Figure 2.12. Dorsolateral view of rat left hemisphere with 1 mm grid (boxes) and cytoarchitectonic borders (heavy lines). Microstimulation map of the rat brain obtained by Neafsey and collegues [Neafsey et al. 1986].

2.7 Summary of the problem analysis

In the previous sections the principal components of a BCI system, with different possibilities to implement them were presented. In order to implement a BCI system, it was necessary to make some choices, strictly dependent on the purpose of the BCI. The animals used for this study were rats, for their anatomical and physiological homologies with human, they were easy to handle during experiments due to their small size and they were inexpensive to obtain and host. Intra-cortical signals were recorded from layer V of pyramidal cells in the primary motor cortex (M1). This area is reported to encode the motor activities and its neurons can generate relatively large amplitude of the extracellular potentials [Moran 2010]. Invasive technique was chosen for the great advantage to have better spatial and temporal resolution, despite the risk of foreign body reaction. Multi-units recordings were used for the feature extraction without spike sorting. Mathematical algorithm were chosen to classify the movement under investigation based on the extracted features. Goal based classification were preferred based on an on-off task. Non linear model were chosen for their adaptability to define complex boundary between data sets. All the data processing was done off-line, no feedback and external device were required. The block diagram 2.13 illustrates the choices made during the previous sections.

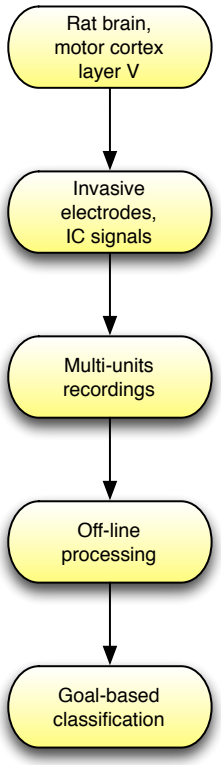

Figure 2.13. The block diagram illustrates the choices made to develop a BCI system for rehabilitation.

Problem Formulation 3

3.1 Problem formulation

To implement an artificial neural network (ANN) and a support vector machine (SVM) classifiers based on intra cortical signals to detect forelimb hit related classes.

3.2 Limitations

3.2.1 Development of a real time algorithm and external control device

As seen in the example 2.4, a BCI system is usually characterized from the type of output that it provides. Although the external device is a fundamental part of the BCI, it is quite difficult to provide a real-time control command which is able to generate correct feedbacks. Such a closed loop system is too complex to be implemented in a short time. For this reason, the data collected were analysed off line with a program implemented in MATLAB (MathWorks, Natick, USA). This gave the opportunity to develop before a robust algorithm to be implemented and test in real time in future experiments. Moreover, it was possible to compare different types of classification and to decide which model to implement in a real-time BCI.

3.3 Solution strategy

In order to investigate whether it is possible from neuronal activity, recorded from the primary motor cortex (M1), to detect the movement, a big amount of data needed to be analysed and classified. The data were collected in a previous experiment and the data processing was performed to denoise the signal and extract the features of interest. In order to detect different classes of paw movement, neuronal firing rates were classified as related to the paw movement or not, in *Hit* and *NoHit* classes respectively.

The project was developed into two parts as shown in the Figure 3.1:

1. *First part* Experimental setup, recording and pre-processing conducted by Sofyan Hammad, PhD student and co-supervisor of the project
2. *Second part* Classification by Martina Corazzol, master student

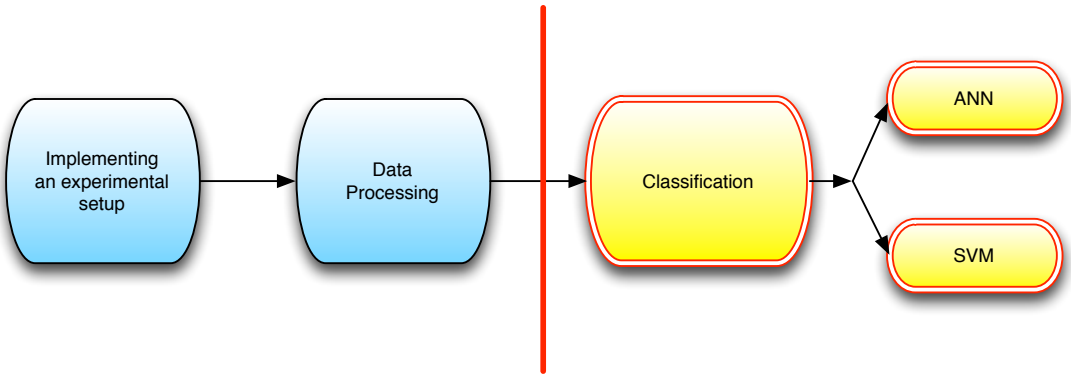

Figure 3.1. Block diagram representing the steps necessary to design a BCI system to be tested in a rat model. On the left of the red line, is shown what was done in a previous experiment to collect and process the data. On the right of the red line, is shown what was accomplished in this project: the classification.

First part

Sprague-Dawely rats were trained to hit a retractable paddle, then a 16 channels electrode were implanted to obtain intra-cortical recordings. The pre-processing included data denoising, spike detection and features extraction.

Second part

Afterwords, the firing rates extracted from the related intra-cortical signals, were used as inputs data for two classification methods: artificial neural network (ANN) and support vector machine (SVM). The first question to be addressed was whether the denoising process with wavelet technique was effective or not to achieve a lower classification error. To answer this question a comparison between the misclassification error rates were done before and after denoising for both the techniques. The second question was whether there were any differences between ANN and SVM classification results.

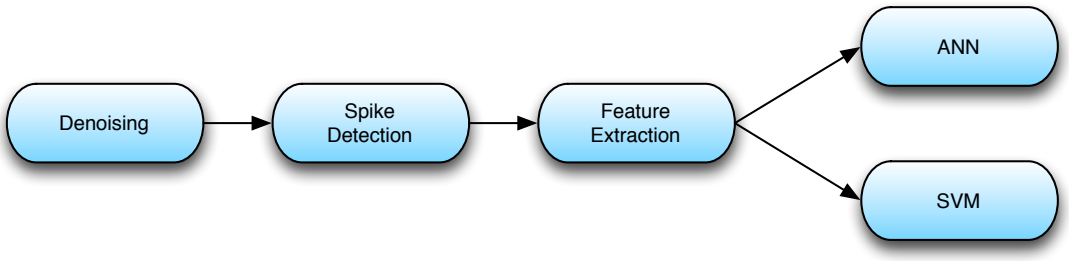

Figure 3.2. Block diagram representing the data processing and the classification. From the raw data the signal was denoised, the spikes detected and the features extracted. Two different classification techniques were applied: ANN and SVM.

Part II

Problem Solution

Experimental Protocol 4

Intra-cortical (IC) signals recorded from the rat motor cortex (M1) were used to decode the brain activity and correlate it with the forelimb movements of the rat. The firing rates of intra-cortical neurons were analysed in order to be classified into two possible classes: *Hit* and *NoHit*.

4.1 Behavioural training

Five adult male Sprague-Dawley rats (470 g ± 30) were included in this research, according to the requirements of the "Danish Committee for the Ethical Use of Animals in Research". During the experiment the rat was placed in an operant conditioning cage equipped with a retractable paddle lever and a food reward deliver mechanism via a pellet dispenser. The rat's task was to hit the response paddle lever three times consecutively with a forepaw in order to obtain the food reward, as shown in Figure 4.1. Three of them preferred to use the left paw, two preferred the right paw. The number of repetitions were chosen to be three to be sure that the rat was doing that specific task, and not a random combination of hits. Only the first hit out of three repetitions was consecutively analysed, because it was free from previous hitting movement artifacts. On the contrary the second and the third hits could also be very near in time to each other, and for this reason be affected by previous hit movements artifacts. It was quite important for the movement classification to analyse only the hits corresponding to a correct paw movement.

The concept of **successful hit** was defined as:

1. The rat hit the response paddle lever three times consecutively
2. The three hits were performed in less than 6 s

The concept of **wrong hit** was defined as:

1. The rat hit before the paddle was ready (paddle lever out of the paddle case)
2. The rat used both paws to hit the paddle
3. The rat hit the paddle with the unexpected paw (according to the preference)

Trials corresponding to the definition of *wrong hit* were discarded from further processing.

4.2 Materials

The following materials were used to conduct the experiment.

TDT, Tucker Davis Technology system

- RX5 Pentusa Base Station
- RA16PA 16-channels medusa pre-amplifier for IC signals
- RA16CH 16-channels chronic headstage for IC signals

Labview VI

A Labview VI was developed to control the paddle position (accessible or not accessible) and the food delivery after three successive hits.

Microelectrode

A 16 channels (4 by 4) home made micro electrode. Made of teflon coated 50 μm tungsten wires expanded in an area of 2 x 2 mm^2.

Paddle lever mechanism

- Lever height 6 cm from the bottom of the behavioural training cage
- Transparent plastic 1 x 7 cm open slit, distant 2 cm from the paddle, to force the rat to stretch the paw
- Metallic mesh, to support the rat in hitting

4.3 Experimental setup

The experimental setup consisted of four main components: the rat cage with the paddle lever and the food reward mechanisms, the TDT system for data acquisition, a digital camera, and a computer. Once the rat hit the lever of the response paddle a digital pulse was sent to the recording system (TDT) to synchronize the neural data with the movement. These digital pulses were simultaneously record by a Labview VI program via data acquisition card (NI USB-6259 BNC, National instrument, USA) with the aim of counting the number of hits and control the related status of the lever (i.e ready for hitting or not ready). After a set of three consecutive hits the paddle lever was automatically retracted for 9 s to allow the rat to eat the reward and to avoid the recording of wrong muscular activity (e.g. chewing). A 16 channels home made tungsten microelectrode array was chronically implanted in the M1 area of the rat brain corresponding to the preferred paw.

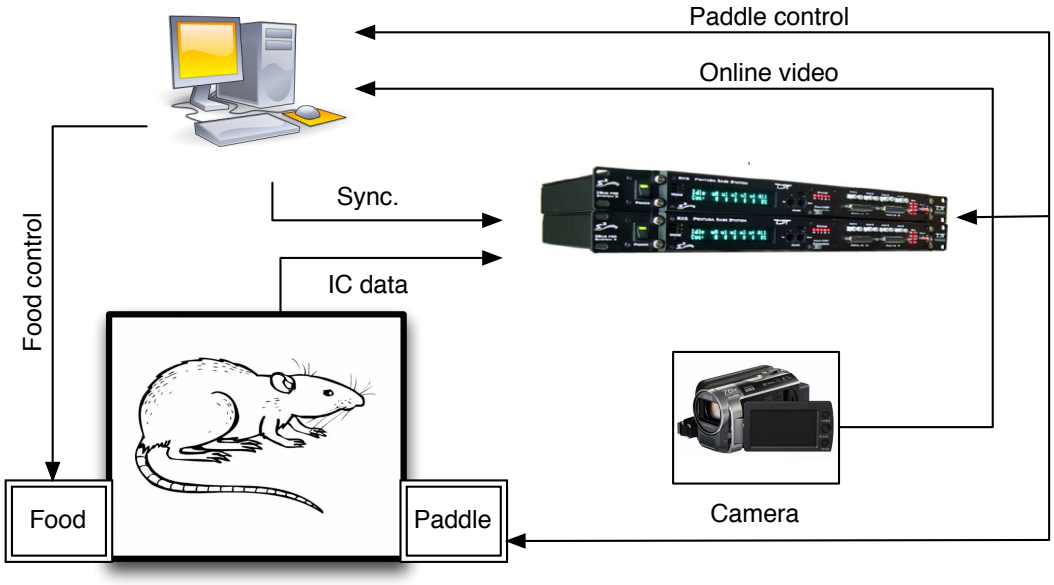

Figure 4.1. Experimental setup scheme: the paddle lever was protracted from the paddle case and ready to be hit by the rat. The IC signals from the rat motor cortex were recorded by the acquisition system (TDT) RX5 Pentusa. Once the lever was depressed by the rat, a pulse signal was sent to the TDT (as Synchronizations) and to the Labview VI program to count the number of hits. After three consecutive hits the Labview VI program sent a control signal to retract the level, a reward delivery signal to the food dispenser and after 9 s sent another control signal to protract the lever again. The experimenter watched the rat during workout via the video camera and also used the Labview VI for sending signals to the neural recording system. These signals marked any trial where the rat used the non-preferred hand or started to hit before the paddle lever was protracted to discard those hitting sets from further analysis.

4.4 Implant procedure

After been anaesthetized with a cocktail of ketamine (100 mg/Kg), xylazine (5 mg/Kg) and acepromazine (2.5 mg/Kg) in doses of 0.1 ml/100 g per body weight, the rat went through a surgical procedure. A craniectomy was performed and the dura was removed in the region of the primary motor cortex (M1) corresponding to the forepaw movement, controlateral to the preferred paw of each rat. A custom made tungsten micro-wire electrode array (Teflon coated, 50 μm, A-M system, Inc, USA) that spanned an area of 2 x 2 mm^2 with 16 channels (4 by 4) was implanted. The implantation area corresponded to 2 mm anterior and 2 mm lateral to the bregma as shown in Figure 4.2. The implant was in a depth of 1.6 mm in order to reach the layer V of pyramidal neurons in the motor cortex, situated in a depth between 1.5 and 2 mm underneath the pia, and responsible for the movement command. Two stainless steel bone screws were mounted on the skull 2-3 mm posterior and lateral to Bregma on both sides. The bone screw on the same side of the electrode was used as recording reference. After the electrode implantation, the exposed brain was covered with collagen-based Gelfoam (Johnson and Johnson, UK) and the electrode was fixed to the skull with dental acrylic (Heraeus, Germany).

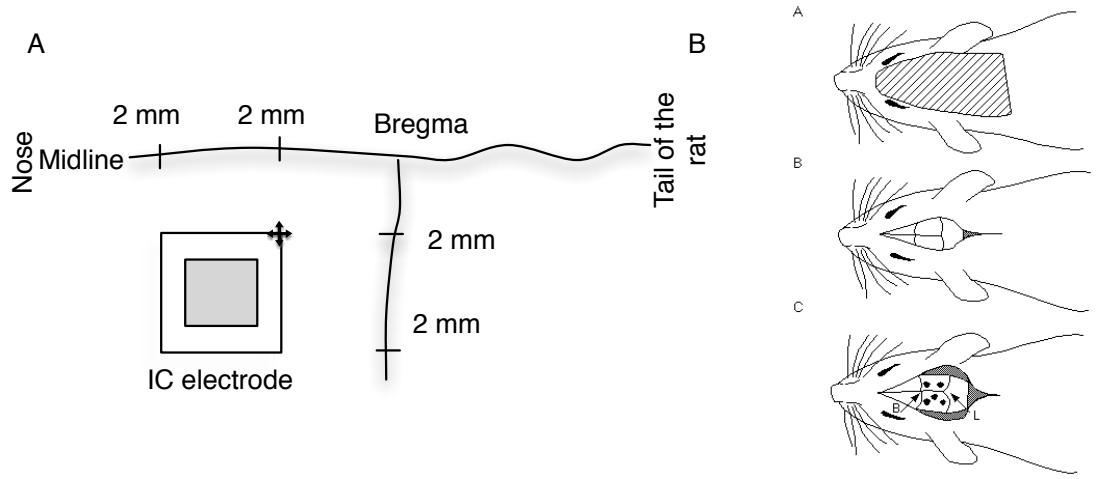

Figure 4.2. Figure (A) shows the position of the implant. The horizontal line represents the midline going through the length of the rat body from the head (left) until the tail (right). The bregma is the crossing point between the coronal and the sagittal suture. Figure (B) shows a pictures of the implant procedure [Dürmüller et al. 2000].

4.5 Recordings

A multi-channels recording system (RX5 Pentusa, Tucker Davis Technology, USA) was used to perform the intra-cortical recordings. The recording system bandpass-filtered the streamed raw data between 0.8 and 8 kHz and sampled it at 24.414 kHz. Furthermore, the paddle response and the control signals were streamed in parallel with the neural data to be used as time stamps for the data analysis.

A total of 34 sessions were recorded from the five rats where 26 sessions (4 out of 5 rats) were analysed. A *recording session* was intended to be a particular day in which recordings were performed for one rat. A recording session contains the *rat task* (hit three times consecutively a paddle lever) repeated for nearly 100 times. Each session is therefore characterized by a certain number of *trials*, or repetitions of the task, varying accordingly to the number of *successful hits* and *wrong hits*, that had to be deleted to prevent artifacts.

4.6 Good channels detection

After the recording sessions an electrical stimulation was applied on the cortical electrode to asses the neural correspondence between the signal and the movement. A monophasic stimulus of 100 Hz, pulse width 200 μm and amplitude 0.1 to 1.5 mA was used. The rat was awake in a prone position with the the paw not supported. The channels corresponding to a forepaw movement were considered as reliable or *good channels*, the other, corresponding to forepaw and neck movements, or no movement, were classified as *not good channel*. The recordings from Rat 5 were excluded because the movement response to the cortical electrical stimulation showed a mixture of paw-neck movement or no response at all channels. Table 4.6 contains the following informations: the *rat name*, the recorded sessions name *N sessions*, the number of correct trials *N trials* and the *good channels*.

Rat name	Session name	N trials	Good channels
Rat 1	08/12/2010	27	[1,3,9,11,15]
	10/12/2010	48	
	13/12/2010	50	
	14/12/2010	56	
	21/12/2010	25	
	23/12/2010	19	
Rat 2	29/11/2010	42	[2,7,11]
	30/11/2010	51	
	06/12/2010	76	
	07/12/2010	60	
	10/12/2010	39	
	22/12/2010	20	
Rat 3	15/12/2010	93	[2,3,6,8,9,10,11,13,14]
	17/12/2010	88	
	20/12/2010	58	
	21/12/2010	88	
	22/12/2010	86	
	23/12/2010	85	
	07/01/2011	84	
Rat 4	15/12/2010	70	[1,3,7,10,13]
	16/12/2010	67	
	17/12/2010	50	
	20/12/2010	44	
	21/12/2010	66	
	23/12/2010	79	
	12/01/2011	70	

Table 4.1. The table presents in the first column the rat name, in the second column the sessions name, in the third column the number of correct trials and in the fourth column the good channels.

Data Analysis 5

5.1 Data processing algorithm

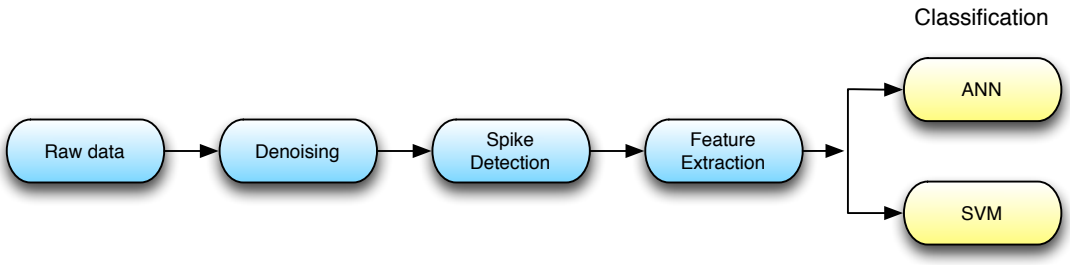

Figure 5.1. Block diagram representing the data processing part and the classification part. From the raw data were applied, in sequence: denoising, spike detection, features extraction and two types of classification.

The data analysis was carried out off-line using MATLAB (MathWorks, Natick, USA). The block diagram in Figure 5.1 shows the data processing and the classification parts. The first part of the data processing, called pre-processing, consisted in: denoising the raw signal with wavelet technique, spikes detection and features extraction, to group and count the spikes into time windows. The second part consisted in the classification realized with two methods: artificial neural network (ANN) and support vector machine (SVM).

5.2 Time windows: Hit and NoHit data

Each session contained a different number of trials, that could vary depending on the number of *wrong hits*. The raw signals were composed by intra-cortical (IC) activity recorded from 16 different channels. For every session and every channel the first hit out of three consecutive hits, composing each *trial*, was considered in the data processing. For every first hit movement a *time window* of 500 ms was extracted, from -400 ms (before the first hit) to +100 ms (after the first hit), where t=0 ms represents the first hit moment, as shown in Figure 5.3. The part of the signal which contained the information about the planning and programming of the forepaw movement was supposed to be before the movement itself, taking place at t=0 ms. The execution of movement occurred after the motor command was processed from the brain, due

to a delay in the propagation between the brain and the the muscles in the forelimb. The data in this time window were referred as *Hit data*. A corresponding window with the same length, 500 ms, were extracted before the *Hit data*, from -900 ms to -400 ms, and was called *NoHit data*. This time window represented a transition time between walking, chewing etc. to hitting. For this reason it was an appropriate window for extracting the *NoHit data*, corresponding to normal brain activity.

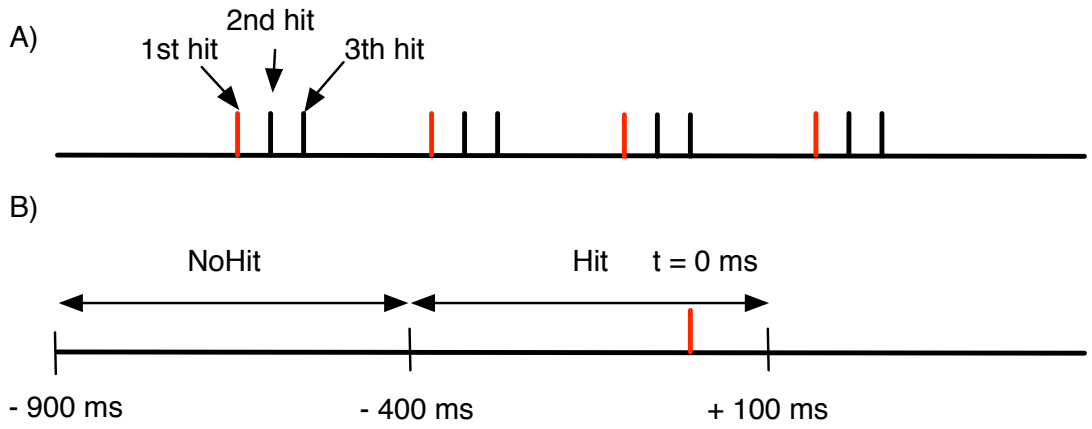

Figure 5.2. A)The figure shows schematically three consecutive hit movements repeated four times. The first hit (red) was considered, the second and the third were deleted. B)The first hit corresponded to t = 0 ms, from that position two time window were selected for the *Hit* and the *NoHit* features extraction .

5.3 Denoising

Wavelet denoising technique was applied in order to denoise the extracted data. The signal was transformed and decomposed into the wavelet domain. A threshold was applied to remove the noise and the resulting signal was reconstructed back to the time domain. The threshold estimation was computed as following :

$$Threshold = \gamma\sigma\sqrt{2ln(N)} \tag{5.1}$$

Where γ is a *threshold correction factor*, σ is the estimate *standard deviation* of the noise, and N is the *number of trials*. Ten different wavelets were applied with 10 different thresholds. The 10 wavelets were selected as following: *Daubechies* 2,4,6, *Coiflets* 2,4,5 *Symlets* 2,4,6 and *Haar*, as shown in Table 5.1. They were used for the similarity between the wavelet shape and a single cell action potential except for haar wavelet which was selected to study the non-similarity effect over the classification. Ten threshold levels were tested by varying the correction factor γ. A set of 100 combinations of wavelets and thresholds were used to compare different denoising results. Table 5.2 presents the values of the 10 *threshold correction factors*, where *threshold n* represents the number of the 10 thresholds.

Table 5.1. Wavelet Table

Wavelet name	**Matlab name**
Daubechies 2	*db2*
Daubechies 4	*db4*
Daubechies 6	*db6*
Coiflets 2	*coif2*
Coiflets 4	*coif4*
Coiflets 6	*coif6*
Symlets 2	*sym2*
Symlets 4	*sym4*
Symlets 6	*sym6*
Haar	*haar*

Table 5.2. Threshold correction factor table

Threshold n	γ
Th 1	0.4
Th 2	0.5
Th 3	0.6
Th 4	0.8
Th 5	1.0
Th 6	1.2
Th 7	1.4
Th 8	1.6
Th 9	1.8
Th 10	2.0

5.4 Spike detection

The potential recorded from a single electrode tip was influenced by the activity of multiple neurons in its vicinity in the layer V of the primary motor cortex (M1). Therefore, the neural information coming from multiple-units was recorded. The spikes were detected along the signal with an adaptive threshold technique, that returned the number of points supra threshold. The threshold level was computed as following:

$$Th_d = 4\sigma_d \tag{5.2}$$

Where:

$$\sigma_d = \frac{median|x|}{0.6745} \tag{5.3}$$

Where Th_d is the *detection threshold level*, σ_d is the estimate of the noise *standard deviation* ad x the bandpass-filtered *data*. A filter was implemented to detect only the first point in each spike, so the count where blocked using a refractory period of 1 ms for minimizing false positive of spikes.

5.5 Feature extraction

After the signal was denoised and the spikes were detected, the firing rate features related to the paw movement were extracted. The features extraction was important since it determined the input of the classifier, and therefore the level of discrimination between different classes. Two possible classes *Hit* and *NoHit* were chosen to classify the paw movement. These two classes corresponded to the correct classification for the features extracted from the *Hit data* window and the *NoHit* data window respectively.

Three features characterizing the *Hit data set* and three features characterizing the *NoHit data set* were extracted from every trial. The features corresponded with the number of spikes counted in bins of 5 ms summed into time window of 120 ms. The three time windows of 120 ms for the *Hit data* were chosen as following:

- Inside the *Hit data* [-400 ms +100ms]
- First interval [-400 ms -280 ms]
- Second interval [-280 ms -160 ms]
- Third interval [-160 ms -40 ms]

and fot the *NoHit data* :

- Inside the *NoHit data* [-900 ms -500 ms]
- First interval [-900 ms -780 ms]
- Second interval [-780 ms -660 ms]
- Third interval [-640 ms -540 ms]

Where t=0 ms corresponds to the first hit time. Figure 5.3 provides a visual example.

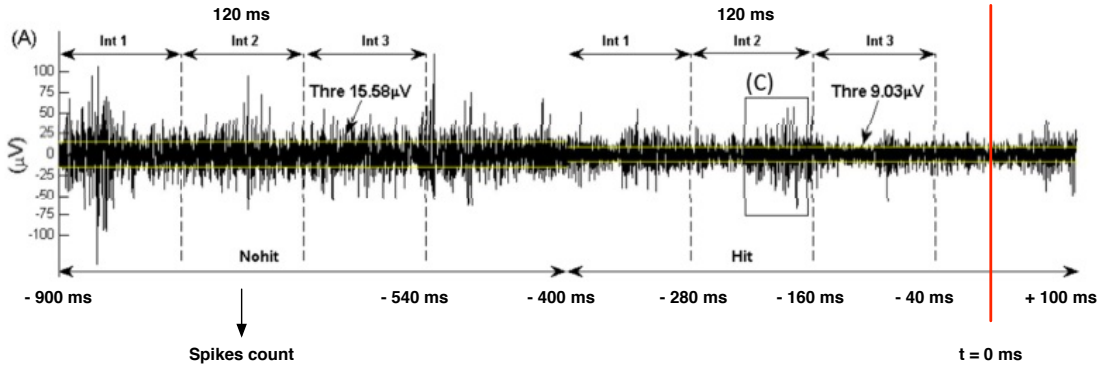

Figure 5.3. The figure shows an example of IC raw signal from a single channel corresponding to an hit movement. The time interval between -900 ms and -400 ms before the hit was called *NoHit*, and the interval between -400ms and +100ms was called *Hit*. Int1, Int2 and Int3 represent the three windows of 120 ms each used to count the spikes and to use them as features in the classification. The yellow line represents two different threshold values calculated independently for *Hit* and *No Hit* data.

This features extraction is supported by a study of Chapin et al. [1999] which reported that neurons connected with activity in the forelimb of the rat began to discharge 30 to 50 ms before forepaws contact. Was also assumed that most of these discharges were stronger before touching than during movements. They discovered that the combined activity of forepaw movement preceded detectable lever movement by 150 ms, and EMG recordings by 100 ms. This evidence justified the time interval chosen for the features extraction.

5.6 Artificial neural network design

After the feature extraction, the classification part was developed. In order to design an artificial neural network (ANN), some choices were made, according to the ANN function. In particular, the main focus was on whether to use supervised or not supervised learning, how many feature to insert in the input layer, the number of hidden layers, the neurons into them and the type of output. About the data set partition it was crucial to decide the percentages of training, testing and validation data, and whether to employ a validation technique to achieve more robust results. Some choices were already made according to the recordings and to the solution strategy, such as the supervised learning, which was independent from the particular technique used but from the feature extraction step. The ANN internal parameters were chosen after a series of pilot experiments conducted in a particular session. This specific session was chosen because in the previous experiment it gave the better classification results: Rat 4 day 16/12/2010.

A supervised learning technique was used to classify the features, since the data set, formed by *Hit* and *NoHit* data, was paired with the correct classification labels. In particular, the data set called *Hit data set* represented the features correlated with a correct movement of the paw. The data set called *NoHit data set* represented the features correlated with a normal brain activity.

The following list presents the basic step followed in order to design an ANN in MATLAB:

1. Collect the data
2. Extract the features of interest (firing rate)
3. Initialize:
   Input vector: a vector containing the features characterizing the data coming from the two classes
   Target vector: a vector containing the correct label for the classification
4. Build the network
5. Divide the data set
6. Train the network
7. Test the network
8. Plot the result

How to collect the data and extract the features has already been described in the paragraphs 4.3 and 5.5.

5.6.1 Build the network

For the purpose of the project, it was necessary to implement a neural network for pattern recognition. A feedforward network was therefore built, with one hidden layer and an output layer. The default input processing functions were *removeconstantrows* and *mapminmax*. For outputs, the default processing functions were also *removeconstantrows* and *mapminmax*. *Mapminmax* normalized inputs/targets to fall in the range [-1, 1] and *removeconstantrows* removed inputs/targets that are constant. [Demuth et al. 1992]

5.6.2 Input vector and target vector

In order to chose the correct number of inputs for the ANN, some pilot experiments were conducted on the features. In particular the *number of features* given as input were quite discriminant for the classification. The input vector in the simplest case was formed by three features, the three time windows extracted from one channel for each trial. Consecutively also the number of neurons in the input layer were three, as shown in Figure 5.6. The Figure presents a two layer network plus an input layer containing the features. The hidden layer and the output layer are characterized by a tansigmoid function, as shown in Figure 5.5. The output layer contained two neurons, that assumed different state between -1 and 1. The choice between the two output neurons (the classes) was made considering the one with the highest value between -1 and 1. To divide the data for the pilot experiments the following partition was chosen: training (70%), validation (10%) and test (20%). The confusion matrix in Figure 5.6 shows the results obtained using only channel 1 for rat 4.

ANN with more than three features

In order to use the information coming from all the 16 channels of the electrode, or at least from the channels related to the movement of the rat forepaw, called *good channels*, more features were considered in the input vector. According with the number of *good channels* the number of features was increased by three for channel considered as "good". In the case of rat 4 the number of *good channels* available were five, therefore 3 features each channel, means that the number of features for the input layer were fifteen. The model presented fifteen features

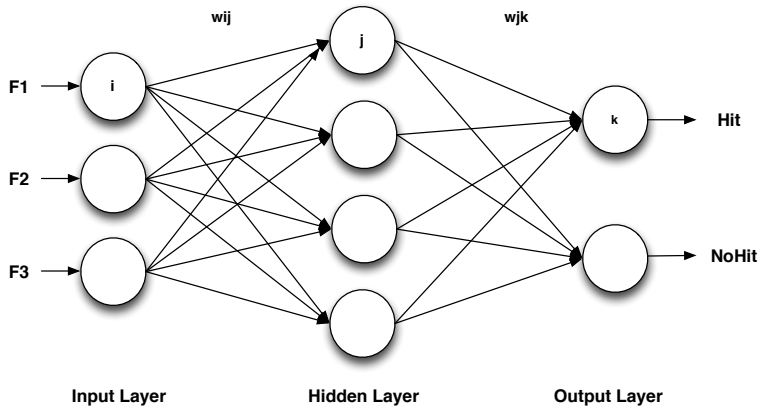

Figure 5.4. Schematic model of a simple neural network implemented with three features in the input layer. Therefore three neurons were required in the input layer, and two, representing the two classes, were required in the output layer.

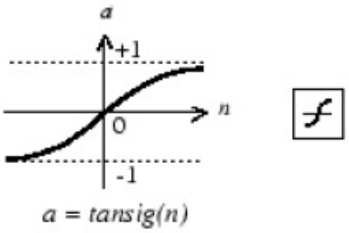

Figure 5.5. Representation of a tansigmoid (*tansig*) function used in the artificial neural network.

in the input layer, ten neurons in the hidden layer and two neurons in the output layer. The correspondent confusion matrix is shown in Figure 5.6. In general, from these results and from what it was experimentally observed, it could be concluded that increasing the number of features the number of misclassified samples decreased. In fact, more features added more information to the classification process. Below is shown how the performance of the neural network behaves adding one channel per time, in a random run without averaging the results.

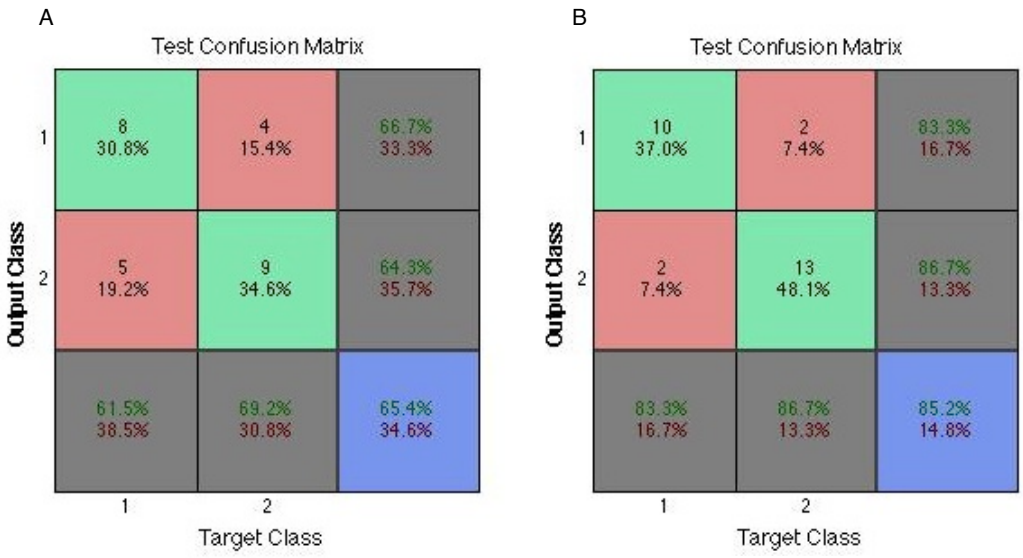

Figure 5.6. Confusion matrix representing the classification results of the *test data*, obtained with three features (A) fifteen features (B) as input vector for the neural network during the pilot experiment.

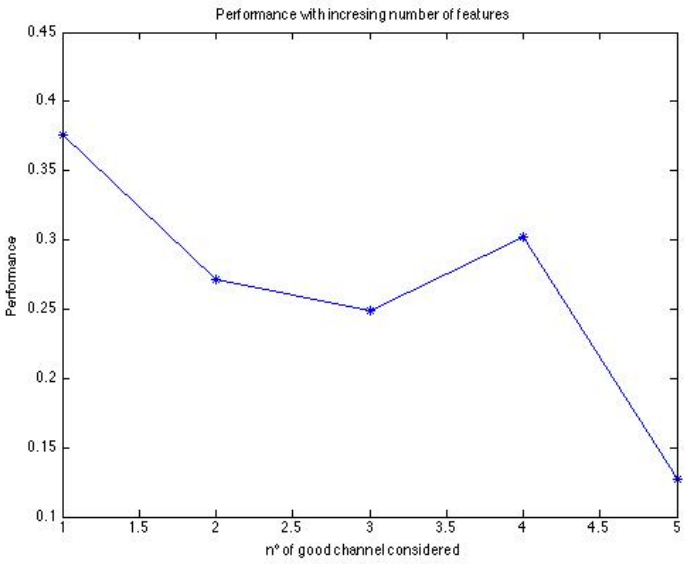

Figure 5.7. The figure shows the how the trend of the performance varied increasing the number of features. The abscissa represents the number of channels used.

Rat name	Session	Number features	Correct hits %
Rat 4	16/12/2010	3	75%
Rat 4	16/12/2010	15	83.3%

Table 5.3. The table shows the rate of correct classification of an hit movement using three features and using fifteen features as input. The first and the second columns represent the rat and the session used, respectively.

Number of neurons in the hidden layer Another important parameter to compute was the number of neurons in the hidden layer. Until now, 10 neurons were used, as a standard number fitting all the applications. After a loop incrementing the number of neurons in the hidden layer from 1 to 30, the following plot was obtained. The circle represents the configuration that achieved overall the best performance.

From this plot, and also from other plots obtained with the partition characterizing the pilot experiments, it could be observed that the performance didn't follow a linear trend of incrementing (decreasing the curve) during the incrementing of the neurons. But it is clear that in the first part, where the number of neurons was small the oscillations were quite bigger compared with the second part of the plot. It could be observed that the optimal number of the neurons was not a fix quantity, but was included between 15 and 30. The solution for the optimal number was a tread off between the number of neurons and the performances. To build the NN, after some trials and some general consideration from the pilot experiment, it was assumed that the optimal number of neurons was 15.

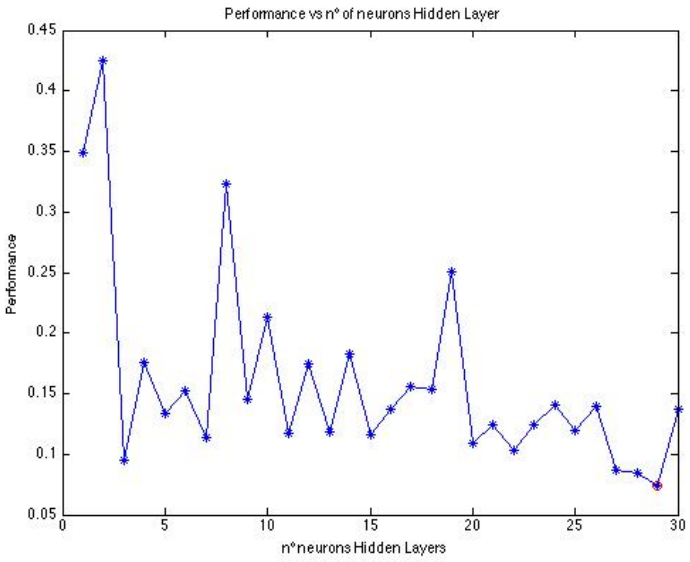

Figure 5.8. The plot shows how the performances varied according to the increment of neurons in the hidden layer.

5.6.3 Dividing the data

In order to train, test and validate the network each *data set*, one for every session, was divided into three sub groups:

Training set which was used to build and train the network, to compute the gradient and to update the network weights and biases. First, the classifier was trained with data for which the correct classification were known. Therefore the training set contained the correct answer.

Validation set which was used to validate the result. Also for this set the correct classification was known but was not used to change the classifier parameters. The validation error normally decreases during the initial phase of training, as does the training set error. However, when the network begins to over fit the data, the error on the validation set typically began to rise. The network weights and biases are selected from the ANN at the minimum of the validation set error.

Testing set which was used to test the trained network, compute the misclassification error rate and to compare different classification methods. In general, if the error on the test set reaches a minimum at a significantly different iteration number than the validation set error, this indicates a poor division of the data set.

Our data set was divided into train (80%) and test (20%). In fact the validation is usually done for every neural network inside the training phase, and therefore the data set to train the neural network was afterword divided into training set and validation set. For the partition it was adopted a standard function, that divides the data set randomly. It was used the neural network tool command *dividerand* was used. For both *Hit* and *NoHit* data sets, the following partition was adopted:

1. Train the NN 80%
2. Test the NN 20%

The *training set* contained 80% of data for training effectively and 20% of data for the validation. Figure 5.9 shows this procedure schematically.

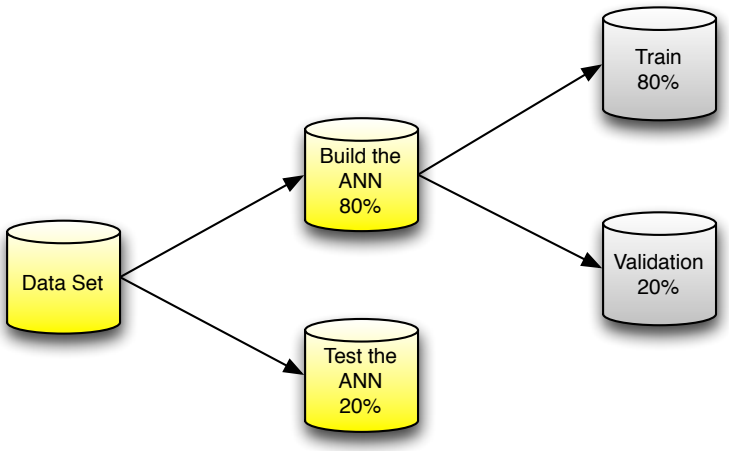

Figure 5.9. The figure represents the partition of the data set: training set (64%), validation set (16%) and testing set (20%).

5.6.4 Training the network

The most widely used method to train a NN is to present like input a series of examples called training set. In particular the output computed is compared with the target value and the weight and bias are modified depending on the error. The training purpose was to tune the values of weights and bias in order to optimize the network performances. The *mean squared error* (*mse*) between network outputs and the targets was used to compute the performance. [Demuth et al. 1992]. The *batch mode* was used to train the network. In the batch mode all the inputs in the training set were applied to the network before the weights were uploaded. These optimization method used either the gradient of the network performance with respect to the network weights, or the Jacobian of the network errors with respect to the weights. They were calculated using a technique that involved back propagation. This technique updated the network weights and biases in the direction in which the performance function decreased most rapidly, the negative of the gradient [Demuth et al. 1992]. The training function used in the feedforword network was the *trainscg Scaled Conjugate Gradient* for pattern recognition.

5.6.5 Testing the network

To avoid possible misleading results deriving from the narrow number of samples, and to make the result more robust, a M-fold cross validation, with M equal to 5 was applied to the data. The procedure were developed and applied to each session as following:

1. Divide randomly the number of trials in 5 equally sized groups
2. Use 5-1 groups to train the system and 1 to test

3. Do this 5 times and average the results

To evaluate the classifier performance of the testing set it was used the *misclassification error rate*. The misclassification error rate, as describe in detail in paragraph 5.11, was defined as the number of misclassified samples divided by the total number of samples in the *testing set*. This procedure was repeated 5 times, the misclassification error rate was computed for every one of the five testing set and averaged in the end. Before starting and after every loop in the 5 fold cross validation the network was reinitialized with the command *init*.

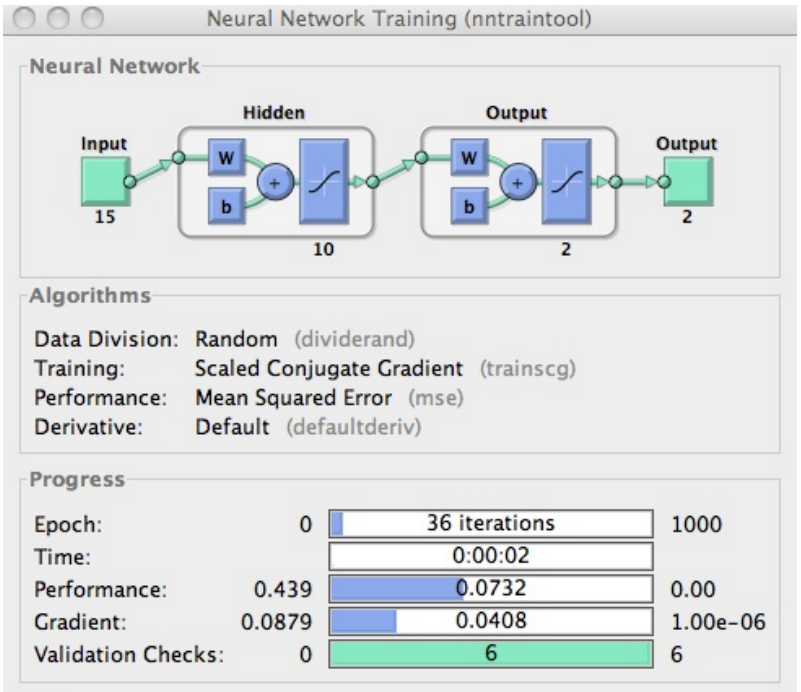

Figure 5.10. Train control window. In the first part are shown the neural network model and the algorithm specifics.

5.7 Summary of ANN design choices

After the pilot experiments conducted in Rat 4 session 16/12/2010, the neural network had the following specifics:

- *Type of network*: Feedforward network for pattern recognition
- *Type of learning*: Supervised learning
- *Number of features in the input vector*: 3 features for each channel, multiplied by the number of good channels
- *Number of neurons in the hidden layer:* 1 hidden layer with 15 neurons inside
- *Number of neurons in the output layer:* 2 output neurons
- *Data partition:* training set (80%) and testing set (20%), with validation embedded in the training phase of the NN taking 20% of the data, random division
- *Training:* Scaled Conjugate Gradient
- *Performance:* Mean Squared Error
- *Validation:* M fold cross validation, with M = 5

Figure 5.11 shows the network performance after adopting the previous specifics.

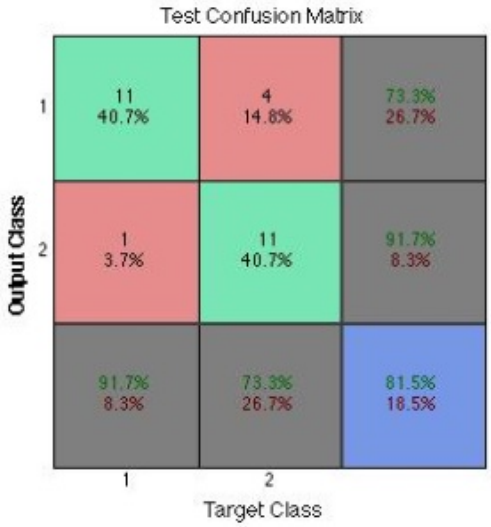

Figure 5.11. Confusion matrix relative to the test data results obtained with 15 input features, 15 neurons in the hidden layer and two output neurons. The confusion matrix was obtained from the pilot experiment, without the 5 fold cross validation.

5.8 Support vector machine classification

The second approach used for the pattern classification was the support vector machin method (SVM). The support vector machine classification algorithm has a different concept to build the decision boundary from the usual linear methods: it does not base the construction of boundaries (e.g. line, curves or hyperplane separating classes) in all data points but only in that ones that are considered critical being nearest to it. So the points that are nearest to the other class became more important and discriminant, as shown in Figure 5.12. The points closer to the boundary are called *support vectors*, and the support vector classifier try to maximize the distance between the critical support vectors to provide a better separation between classes. Finding the boundary that maximizes the margin is a classical optimization problem.

Input and target vectors

The same considerations for the classification using the ANN is still valid in the support vector machine technique, which also performed a *supervised learning*. The same partition strategy was used for the SVM, giving the same features as input for the classification and the same target vector. In fact, three features, corresponding with three time windows, for each channel were used like input. The number of inputs varied dealing with the session and according with the number of *good channel* found. In the target vector 0 was assigned as a label to the *Hit* data and 1 to the *Nohit* data.

Partition and test

To validate the results a M fold cross validation was performed, with M equal to 5. The data set was divided into 5 groups, the first time, four were used for training (80%) and one for testing(20%).

1. Train the NN 80%

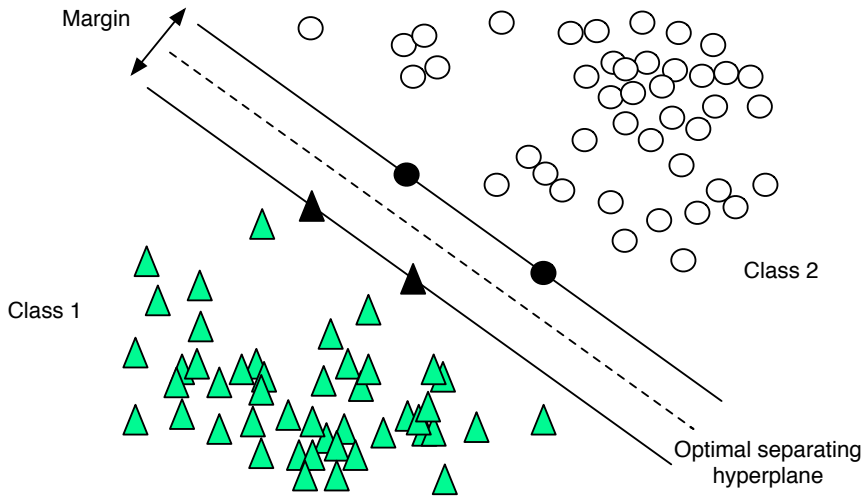

Figure 5.12. The points closest to the other classes are more critical in separating the classes and are called support vectors.

2. Test the NN 20%

In the second loop another combination of four was chosen and so on for five times. After the fifth loop a mean between the results was done, and the mean misclassification error was found.

Transformation

If the two classes are not *linearly separable* two possible solutions can be implemented: assume that they are and accept the error or transform the data into higher dimensions and than apply the transformation. The second solution was implemented and a *polynomial kernel* was used. The method to create the hyperplane was chosen to be the Sequential Minimal Optimization (SMO).

5.9 Summary of the SVM design choices

- *Type of learning*: Supervised learning
- *Number of features in the input vector*: 3 features for each channel, multiplied by the number of good channels
- *Data partition* : training set (80%) and testing set (20%)
- *Training*: Sequential Minimal Optimization
- *Transformation function*: Polynomial kernel
- *Validation*: M fold cross validation, with M = 5

5.10 Evaluating the classifier performance

Perfect classification never occurs, in fact a classification error can occur if classes overlap. There are several methods for quantifying the classification errors, among these can be found the *confusion matrix* and in particular *misclassification error rate* .[Semmlow 2009a]

In this project the confusion matrix was used to quantify the results of the classification. The confusion matrix gave the number of correct and incorrect samples classifications, usually as percentage. It is formed by 2 rows and 2 columns, which represented respectively the 'input', the true result from the classification, and the output, the result suggest form the classifier. The principal diagonal presented the data with a correct classification, on contrary, in the other diagonal two squares represented the misclassified data. In the classification task there were two possible types of errors:

Type I *Incorrectly recognized samples,* also called False Positive (FP), which were **Hit** classified as **Nohit**

Type II *Not recognized sample,* also called False Negative (FN), which were **Nohit** classified as **Hit**

True and false are adjective referring to the correct or incorrect classification, positive and negative refer instead to the class to which the sample is assigned after the classification. Positive

was assigned to the hit movements. The total number of errors made by the classifier was called misclassified samples, and defined as:

$$Misclass\ Samples = FP + FN \tag{5.4}$$

	Hit	*NoHit*
Hit	TP	FP
NoHit	FN	TN

Table 5.4. General scheme of the confusion matrix for the classification into two possible classes : *Hit* and *NoHit*.

5.11 Misclassification error rate

To evaluate the performance of the two classifiers and compare them, the misclassification error was taken into account. The misclassification error was defined as the number of misclassified sample overall and computed as:

$$MisClass\ Error\ Rate = 100\ \frac{MisClass\ Samples}{Test\ Samples}\ \% \tag{5.5}$$

Where the misclassified samples have already been defined 5.4 as those samples that were incorrectly recognized or not recognized. The *Test Samples* was the total number of samples used for testing the classifier.

Data without denoising

For every rat and every session, 16 channels were available. Each session was characterized by a certain number of trials and each trial by 3 features for each channel (spikes count in three consecutive time windows). These features were used as input for the classification, giving like output two possible classes: *Hit* and *NoHit*. Therefore, every session is characterized by one misclassification error.

Denoised data

As mentioned before, in the pre-processing, wavelet denoising techniques were used. In particular, the combination of 10 wavelets and 10 thresholds was systematically applied to every rat and every session, resulting in 100 different denoising solutions for the denoised data. In order to decide which combination, out of the 100, was the most effective for every session, the classification was applied to all the possible denoised features and the misclassification error rate was computed. In this way 26 different tables, corresponding to 26 sessions, with 100 elements each were obtained. For every session, and every 100 elements tables, the minimum, maximum, mean and standard deviation were extracted, e.g Figure D.1. To present the best

misclassification error rate, and the respective pair (wavelet, threshold) of denoising, the pairs were stored in a Pair Table, e. g Figure D.2. After doing this for each session all the couples that gave the best misclassification error rate for every session were summed in a Pair Table valid for all rats. The results are presented in the next chapters.

Results 6

This chapter presents the results obtained using ANN and SVM classifiers. To verify the effect of the denoising algorithms a comparison between classification of denoised features and features obtained without denoising is proposed for both methods. Afterwords, ANN and SVM misclassification error rates are compared to analyse the differences between the two classification methods.

6.1 Classify denoised and not denoised data with ANN

Features extracted from denoised and not denoised data were classified using ANN. The misclassification error rates were calculated in both cases. To evaluate the effect of the denoising the two misclassification error rates were then compared. The misclassification error rate in the case of not denoised data was simply the misclassification error rate obtained from the features extracted from raw data. The misclassification error rate in the case of denoised data represented the minimum misclassification error rate obtained out of 100 different denoising combination of wavelets and thresholds. After the minimum misclassification error rates were extracted for each one of the 26 sessions, for the denoised and not denoised data, they were plotted together for each session. The Figure 6.1 shows the results for the four rats.

From the plots it can be observed that the denoised signal (red) provided better features for the classification, in fact they corresponded to a lower misclassification error rate. To demonstrate it statistically a paired student test (t-test) was applied to decide if the two populations of data (misclassification error rate before and after denoising) were significantly different. The t-test assesses whether the means of two groups are statistically different from each other. Table 6.1 shows that the p-value calculated for all the rats was lower than 0.05. It can be concluded that with a confidence of 95% the two groups represented statistically different populations. Therefore applying a denoising technique made a difference in the results, providing a lower misclassification error rate for the ANN.

It was interesting to examine which combinations of wavelets and thresholds provided the best classification. The minimum misclassification error rate was extracted for each session, out 100 denoising combinations (wavelet and threshold). The pairs resulting more times to achieve the better classification over the 26 sessions were stored in the Pair Table 6.2. The identified pairs resulted to give a better classification 2 times out of 26 sessions.

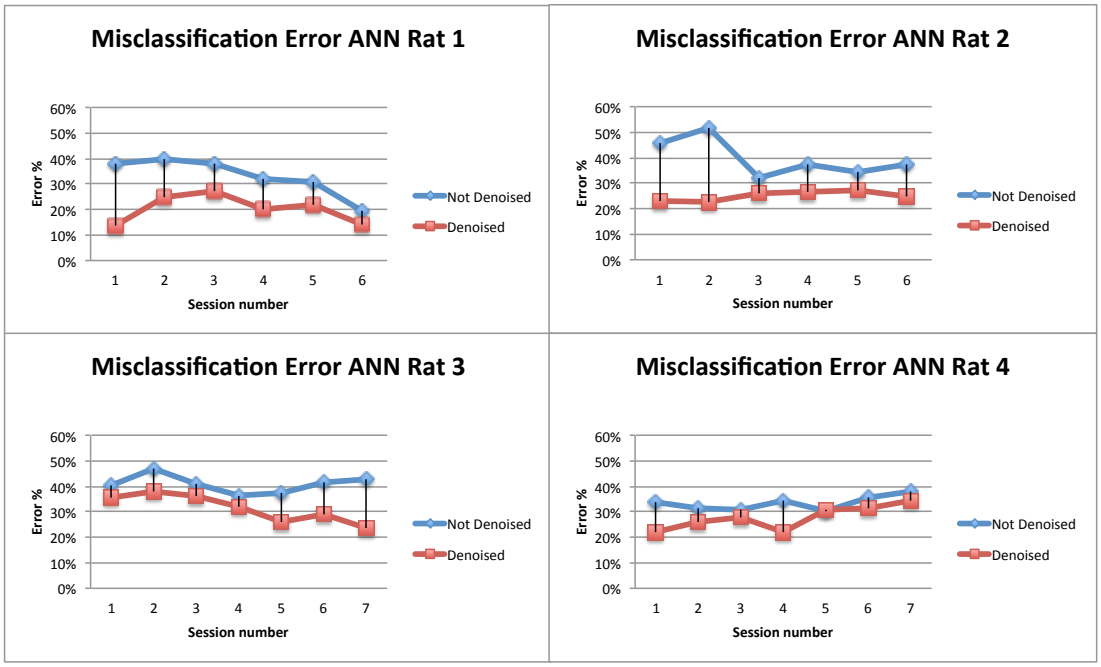

Figure 6.1. The plots show the misclassification error rate obtained classifying features extracted from the not denoised data (blue), and the misclassification error rate obtained classifying features extracted from the denoised data (red), using in both cases ANN classifiers. The abscissa presents the number of session for each rats.

TABLE 6.1
P-Value Denoised and Not Denoised Data ANN

Rat name	**p-value**	**data processing**	**average ± sd %**
Rat 1	0.005	not denoised	33.0 ± 7.6
		denoised	20.3 ± 5.4
Rat 2	0.011	not denoised	39.8 ± 7.4
		denoised	25.2 ± 1.9
Rat 3	0.004	not denoised	41.0 ± 3.6
		denoised	31.5 ± 5.5
Rat 4	0.018	not denoised	33.4 ± 2.8
		denoised	27.8 ± 4.7

Table 6.1. The table presents the p-values obtained analysing the misclassification error rates for not denoised and denoised data. The third and fourth columns report the average and the standard deviation of the values for each rat.

	Wavelet name									
Th	*db2*	*db4*	*db6*	*coif2*	*coif4*	*coif6*	*sym2*	*sym4*	*sym6*	*haar*
0,4	0	0	0	0	0	0	1	0	1	1
0,5	0	0	0	0	1	0	1	0	0	0
0,6	0	0	0	1	1	1	1	0	0	1
0,8	0	0	0	0	0	0	0	0	2	0
1,0	0	2	0	0	1	0	0	0	1	1
1,2	0	0	0	0	0	0	0	1	0	0
1,4	0	0	1	1	1	0	0	0	0	0
1,6	0	0	0	0	0	0	0	0	0	0
1,8	0	0	2	1	0	1	0	0	2	0
2,0	0	1	1	0	2	0	0	0	0	0

Figure 6.2. **Pairs of wavelets and thresholds for ANN.** The table shows how often, out of the 26 sessions, a certain denoising pair resulted to provide the lower misclassification error rate, out of the 100 possible combinations.

6.2 Classify denoised and not denoised data with SVM

The same technique was used to determine if also with the support vector machine classifier the denoising algorithm produced an increasing of the classifier performance and therefore a decreasing of the misclassification error rate. The misclassification error rates obtained classifying features extracted from raw data (without denoising) were compared with the misclassification error rates obtained classifying the features extracted from denoised data in Figure 6.3. Every point in the graph referring to denoised data (red) represents the minimum misclassification error rate extracted from 100 different denoising pairs (wavelet and threshold). Every point referring to the data without denoising (blue) represents, instead, the misclassification error rate computed for each session.

It can be observed that the denoised signal always provided a better features for the classification. In fact the misclassification error rate referring to denoised features was in all cases lower. A paired t-test was performed to determine if the denoising was effective. As shown in Table 6.2, the p-value for all the rats were lower than 0.05 ($p<0.05$). It can be concluded that the difference between the two misclassification error rates was significant, and that the denoised data provided better classification in all cases.

For every rat the minimum misclassification error rates computed from denoised data corresponded to the configuration of denoising (denoising pair) that allowed the result. A table similar to the previous one was obtained, Table 6.4. The pairs that allowed a lower misclassification error rate were : *(db2 0.5), (coif2 0.8), (sym2 0.5)* and *(sym6 0.6)*.

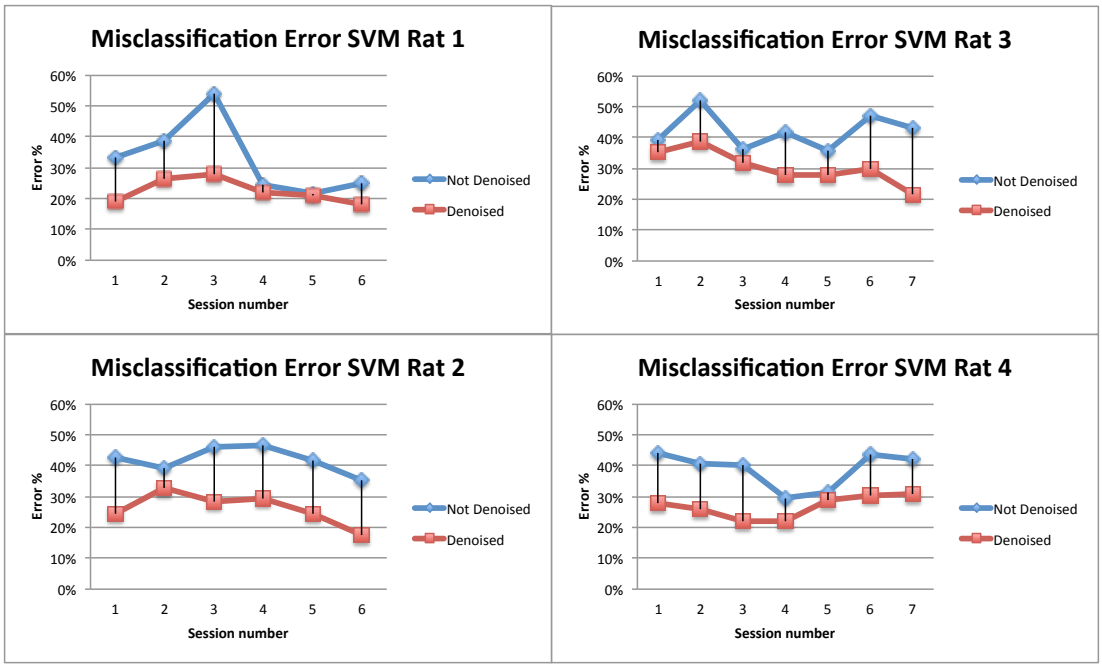

Figure 6.3. The plots show the misclassification error rates obtained classifying features extracted from the not denoised (blue) data and the misclassification error rates obtained classifying features extracted from denoised data (red), using in both cases SVM classifiers.

TABLE 6.2
P-Value Denoised and Not Denoised Data SVM

Rat name	**p-value**	**pre-processing**	**average ± sd %**
Rat 1	0.038	not denoised	32.9 ± 12.1
		denoised	22.3 ± 4.0
Rat 2	0.011	not denoised	41.9 ± 4.4
		denoised	26.1 ± 5.2
Rat 3	0.003	not denoised	42.0 ± 5.8
		denoised	30.4 ± 5.5
Rat 4	0.001	not denoised	38.7 ± 6.0
		denoised	26.8 ± 3.6

Table 6.2. The table presents the p-values obtained analysing the misclassification error rates for not denoised and denoised features. The third and fourth columns report the average and the standard deviation of the values for each rat.

	Wavelet name									
Th	*db2*	*db4*	*db6*	*coif2*	*coif4*	*coif6*	*sym2*	*sym4*	*sym6*	*haar*
0,4	1	0	0	0	0	1	1	0	0	0
0,5	2	0	0	0	1	0	2	0	1	0
0,6	1	1	0	1	0	0	1	1	2	0
0,8	0	0	0	2	0	0	0	1	0	0
1,0	1	0	1	0	0	0	1	0	1	1
1,2	1	1	1	1	1	0	1	0	1	0
1,4	0	0	0	0	0	0	0	1	0	0
1,6	0	0	0	0	0	0	0	0	0	1
1,8	0	1	0	0	0	0	0	0	1	0
2,0	0	0	0	0	0	0	0	0	0	0

Figure 6.4. **Pairs of wavelets and thresholds for ANN.** The table shows how often, out of the 26 sessions, a certain denoising pair resulted to provide the lower misclassification error rate, out of the 100 possible combinations.

6.3 Comparison between ANN and SVM classifiers

After the usefulness of the denoising algorithm was demonstrated, it was interesting to investigate whether the results were changing using two different types of classification: artificial neural network (ANN) and support vector machine (SVM). The two misclassification error rates, obtained after the denoising of the raw data, were compared and plotted in the same graph for all the rats, Figure 6.5.

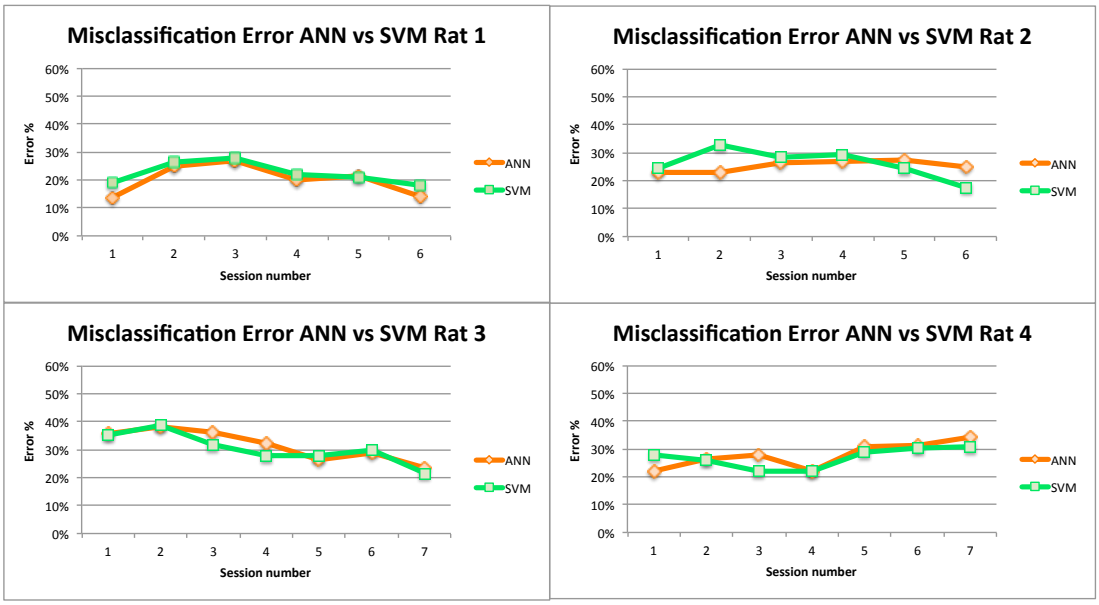

Figure 6.5. The plots show the misclassification error rate obtained with a ANN classifier (blue) and the misclassification error rate obtained with a SVM (red).

The paired t test revealed in all cases there were no substantial difference (p>0.05) between the distributions of the two misclassification error rates, as shown in Table 6.3. For this reason the two methods were considered comparable.

TABLE 6.3
P-Value ANN vs SVM

Rat name	**p-value**	**pre-processing**	**average ± sd %**
Rat 1	0.067	ANN	20.3 ± 5.4
		SVM	22.3 ± 4.0
Rat 2	0.720	ANN	25.2 ± 0.0
		SVM	26.1 ± 0.3
Rat 3	0.293	ANN	32 ± 5.5
		SVM	30.4 ± 5.5
Rat 4	0.476	ANN	27.8 ± 4.7
		SVM	26.8 ± 3.6

Table 6.3. The table presents the p-values obtained analysing the population misclassification error rate for denoised and not denoised features. The third and fourth columns report the average and the standard deviation of the values for each rat.

Discussion 7

7.1 Methodological considerations

The rat as a model

For long time, the rat has been the animal of choice to study neurological diseases. In fact, the brain size allowed surgical procedure and chronical neural recordings with relative ease [LeDoux 2005, chap. A2]. It was demonstrated that the rat forepaw movements exhibit anatomical similarities with human shoulder mechanisms. Also other reach-to-grasp rat movements pointed out the similarity with movement control in human. Furthermore, it was recently established that rats are capable of some independent control of the digits. Other studies about movement related encode information used monkeys [Ryu & Shenoy 2009], which are genetically closest to humans. Monkeys experiments aim is usually to predict the trajectory, velocity and acceleration of hand movements [Wessberg et al. 2000]. Despite this, monkeys are more expensive and difficult to manage. Moreover, the choice to use the rat as a model was consistent with the general aim of the experiment: to detect the paw movements during an on-off task. [Hyland & Jordan 1997]

Implant site and number of channels

The layer V in the primary motor cortex (M1) is related to the paw movements and therefore it reflected the appropriate site for the recordings. The same area was used also by Chapin et al. [1999]. In previous experiments, electrodes with 8 to 16 channels were used for rats by Olson et al. [2005], and from 16 up to 32 for monkeys by Wessberg et al. [2000] The microstimulation was performed to correlate the implant site with specific response in the body and therefore to asses the number of channels reaching the target area. Different rats had different channels connected with the neurons related to the paw movements. It was observed that the number of good channels was not proportionally related to the misclassification error rate in different rats. In fact the misclassification error rate depended also from other factors, such as the level of the noise in the signal and the number of trials. However, it was noticed that increasing the number of channels within the same session corresponded with an increasing of the performances. Consecutively, future experiments will not require to increase the number of channel (with the risk to record from a too broad area) rather than to increment the precision of the channel targets.

7.2 Data Analysis

The signal processing was divided into four parts: denoising, spikes detection, features extraction and classification. The denoising was effective and incremented the results of each type of classification. In fact the paired t test confirmed that the two misclassification error rates, coming from not denoised and denoised data, belonged to different statistical distribution. Based on the average result, the denoised data provided a better misclassification error rates.

In this study no spikes sorting were performed. On the contrary, in some previous studies the spiking activity from single neurons was extracted. This involved complex signal processing to identify individual cell and separate them accordingly to different waveforms. It was estimated that from ten to hundred neurons were necessary to obtain an accurate performance [Wessberg et al. 2000][Taylor et al. 2002][Carmena et al. 2003]. This procedure requires time, power supply and an human intervention to control the ongoing waveform on each channel at the beginning of each session [Fraser et al. 2009]. Furthermore, the decision to avoid the spike sorting is supported by Stark & Abeles [2007], who reported that multi-units activity (MUA) provided an accurate prediction of the upcoming movement. MUA recordings were obtained more easily than single units activity (SUA) and were stable over time. Compared with local field potential (LFP), MUA were less redundant, and compared with other intra-cortical signal they were more informative. In their articles Stark & Abeles [2007] demonstrated also that predictions based on multichannel MUA were superior to those ones based on spikes and LFP. For this reason the choice to use multi-neuronal activity was not a simplification, but a way to investigate the information coming from an entire population.

Rat task and classification

The rat task consisted on hitting a retractable paddle three times consecutively. A two classes classifier was employed to detect the paw movement. This classifier did not give a really accurate description of the movement, but allowed the movement detection. This decoding technique can be applied in a real word to detect the intention to move the arm or in the communication prosthesis to achieve a certain goal, without predicting continuous kinematic parameters. This reflects the design of the BCI, implemented to decode an on-off task translated into two possible classes. Therefore, this kind of classification can be employed in all the applications requiring binary outputs. Olson et al. [2005] used a small number of motor cortical signals (8-10 single units) from rats and a support vector machine to detect "left" and "right" movements. In this way the simple pressing task could mimicking the control of a wheelchair turning left and right, or move a computer cursor to select different items. A different application was developed by Chapin et al. [1999] where rats obtained a water reward by depressing a spring-loaded lever which proportionally presses a robot arm to a water dropper. This continuous movements allowed mathematical transformations to predict the lever trajectory and to implement also motor prosthesis.

7.3 Results

The results of the classification showed that the two methods, ANN and SVM, were comparable. This type of task was classified erroneously in average in the 14% of the cases (in the best case) and in the 38% of the cases (in the worst case) with ANN, and in the 18% and 39% with SVM, respectively. Both of them performed non linear model, capable of solving paradigms that linear computing cannot. The SVM was a classification algorithm fitting this problem because it allowed to develop a robust classifier using only a few free parameters even for high-dimensional data. In fact, the evaluation of the decision function at the heart of the SVM is based on the evaluation of a subset of the training data (the *support vectors*). SVM were successfully applied in many fields resulting in improvement over traditional methods. Olson et al. [2005] achieved an average of accuracy of 87% with a leave-one-out cross validation and 85,5% with an online control. This accuracy was better than the one predicted by bayesian methods.

Neural networks are a popular method that generate complex decision boundaries. Wessberg et al. [2000] applied ANN for fitting parameters without finding appreciable differences between linear and ANN methods. Nicolelis et al. [1998] used backpropagation artificial neural network to investigate the representation of tactile information in three areas of the primate somatosensory cortex. They assessed that ANN technique was widely used for non-linear pattern recognition, not only in computer science and engineering but also in neuroscience. The feedforward backpropagation (BP) ANN was used for a supervised learning technique that produced parametric outputs in probabilistic ranges.

7.4 BCI improvements

Future developments of the experimental protocol imply some implementations regarding a real time interface, to be develop in a real word application. Possible implementations are listed below, and feature prospective are discussed in next session.
Improvements related to this experiment:

- Features related with the amplitude of the spikes can be extracted
- Relationship between the three time window can be deepen, and other solution to extract firing rate features can be tested
- Other pattern recognition methods (e.g linear methods) should be investigated to verify if significant changes will happen in the misclassification error

Improvement related to a real time experiments:

- Some real time prediction with ANN and SVM should be performed, training ANN with the sessions recorded for the same rat
- The data processing should be reduced to the very first part of the experiment to find the appropriate denoising pair for every rat and every session and allow the application of the algorithm online
- A visual feedback related to the neural activity should be provided
- Try a brain control mood

7.5 Future prospectives

Implementation and translation into real world

In order to implement such an interface into real time application, the signal processing time should be decrease. In fact the algorithm was developed to evaluate the denoising technique and to provide a wide range of denoising possibilities. In a real time experiment the first 50 trials should be used for extracting the most effective denoising combination, that should then be used for the experiment duration. The two algorithms would perform very differently in a real time application. In fact the data obtained referred to a validation technique which runs the code for 5 times, and extracts a mean misclassification value. For making prediction one combination of wavelet and threshold should be chosen and only one technique should be applied. In this case the first part of the data would be used for training the network, and the second part to predict the paw movements. ANN and SVM were already used as a real time algorithm for trajectory prediction [Wessberg et al. 2000], therefore they can also be applied in a future real time experiment.

Feedback and plasticity

A necessary improvement to develop the experiment and translate it into a real world context consists in providing feedback related to the neural activity [Taylor et al. 2002]. In fact previous experiments on monkeys demonstrated that the cells tuning properties changed when a brain-controlled movement was used. This was realized developing a closed-loop system between the brain and the hypothetical prosthetic limb. Feedback information helps to learn to use the brain activity to interact with a prosthesis or with a computer cursor. Studies in rats [Chapin et al. 1999] revealed that after a visual feedback information and a proper reward they progressively ceased to produce limb movements. This indicated that motor cortical signals can be generated by cortical neurons without any muscles activity. This leads to the hypothesis that a closed-loop control with an external device could incorporate electronic, mechanical and virtual representation in the brain and use them as extension of the body. The plasticity of the brain, proved by the cortical reorganization after injuries, supports this theory. [Nicolelis et al. 2001].

Conclusion 8

Brain computer interfaces (BCI) system aims to restore basic motor functions or to assist partially paralysed or completely paralysed patients. Recent studies showed that motor parameters, such as limb trajectory and cognitive parameters, such as the goal of an action, can be predicted by decoding motor related intra cortical signals from the primary motor cortex. In fact, mathematical model can be used to translate intra-cortical (IC) signals into features related to particular hand movements. The aim of the project was to investigate different decoding algorithms in order to investigate whether IC signals can distinguish a specific hit movement from the normal brain activity during an hitting task.

Sprague-Dawley rats was trained to hit a retractable paddle lever in an operant conditioning cage for three times consecutively. An electrode was placed within the topographical representation of the paw movement in the motor cortex, verified by visual inspection and electrical stimulation. IC signals were obtain in 26 sessions while the rat was performing the hitting task. The signal was first denoised with 100 different combinations of wavelets and thresholds. The firing rate features were extracted after the spikes detection and used as inputs for two types of non-linear classifier: artificial neural network and support vector machine.

The results verified the denoising effectiveness and allowed to find some denoising pairs (wavelet and threshold) which gave better classification results for each rat and session. The two methods provided good misclassification error rates ranging from 14% to 38% for ANN, and 18% to 39% for SVM. Applying a paired t test resulted that there was not statistical differences between the two groups of errors. Therefore ANN and SVM provides compareble misclassification error rates. Despite this observation ANN achieved the best mean misclassification error rate.

The rat model was an appropriate choice related to the problem formulation. In fact it allowed the detection of the hit movement in an on-off task. The classifier can be implemented in a real world applications that require binary decision responses or an application that can detect the intention to move.

Bibliography

R. Andersen, et al. (2004). 'Selecting the signals for a brain-machine interface'. Current opinion in neurobiology **1**4(6):720–726.

N. Birbaumer (2006). 'Breaking the silence: brain–computer interfaces (BCI) for communication and motor control'. Psychophysiology **4**3(6):517–532.

J. Carmena, et al. (2003). 'Learning to control a brain–machine interface for reaching and grasping by primates'. PLoS biology **1**(2):e42.

J. Chapin, et al. (1999). 'Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex'. Nature neuroscience **2**:664–670.

H. Demuth, et al. (1992). MATLAB: Neural Network Toolbox: User's Guide. The Math Works.

M. E. Donna L. Hudson (1992). Neural network and artificial intelligence for biomedical engineering.

R. Duda, et al. (2001). 'Pattern classification'. New York: John Wiley, Section **1**0:l.

N. Dürmüller, et al. (2000). 'Vigilance-Controlled Quantified EEG in Safety Pharmacology'. Current Protocols in Pharmacology .

E. Evarts, et al. (1984). 'Neurophysiological approaches to higher brain functions' .

E. Fetz (1969). 'Operant conditioning of cortical unit activity'. Science **1**63(3870):955–958.

G. Fraser, et al. (2009). 'Control of a brain–computer interface without spike sorting'. Journal of neural engineering **6**:055004.

A. Georgopoulos, et al. (1986). 'Neuronal population coding of movement direction'. Science **2**33(4771):1416–1419.

A. Ghazanfar, et al. (2000). 'Encoding of tactile stimulus location by somatosensory thalamocortical ensembles'. The Journal of Neuroscience **2**0(10):3761–3775.

D. Hebb (1949). 'The organization of behavior: A neuropsychological theory'. New York .

B. Hyland & V. Jordan (1997). 'Muscle activity during forelimb reaching movements in rats'. Behavioural brain research **8**5(2):175–186.

E. Kandel, et al. (2000). Principles of neural science, vol. 4. McGraw-Hill New York.

P. Kennedy & R. Bakay (1998). 'Restoration of neural output from a paralyzed patient by a direct brain connection'. Neuroreport **9**(8):1707.

B. Kolb & R. Tees (1990). The cerebral cortex of the rat. MIT press Cambridge, MA:.

M. Lebedev & M. Nicolelis (2006). 'Brain-machine interfaces: past, present and future'. TRENDS in Neurosciences **2**9(9):536–546.

M. LeDoux (2005). Animal models of movement disorders, vol. 1. Academic Press.

J. Lewis (2004). 'A short SVM (support vector machine) tutorial'. CGIT Lab/IMSC, University Southern California .

P. Limousin & I. Martinez-Torres (2008). 'Deep brain stimulation for Parkinson's disease'. Neurotherapeutics **5**(2):309–319.

M. Lotze & U. Halsband (2006). 'Motor imagery'. Journal of Physiology-Paris **9**9(4):386–395.

S. Mason & G. Birch (2003). 'A general framework for brain-computer interface design'. Neural Systems and Rehabilitation Engineering, IEEE Transactions on **1**1(1):70–85.

D. Moran (2010). 'Evolution of brain–computer interface: action potentials, local field potentials and electrocorticograms'. Current opinion in neurobiology **2**0(6):741–745.

E. Neafsey, et al. (1986). 'The organization of the rat motor cortex: a microstimulation mapping study'. Brain Research Reviews **1**1(1):77–96.

M. Nicolelis et al. (2001). 'Actions from thoughts'. Nature **4**09(6818):403–408.

M. Nicolelis, et al. (1998). 'Simultaneous encoding of tactile information by three primate cortical areas'. Nature neuroscience **1**:621–630.

M. Nicolelis & M. Lebedev (2009). 'Principles of neural ensemble physiology underlying the operation of brain–machine interfaces'. Nature Reviews Neuroscience **1**0(7):530–540.

A. Nobunaga, et al. (1999). 'Recent demographic and injury trends in people served by the Model Spinal Cord Injury Care Systems'. Archives of physical medicine and rehabilitation **8**0(11):1372–1382.

R. Nudo (2007). 'Postinfarct cortical plasticity and behavioral recovery'. Stroke **3**8(2):840–845.

B. Olson, et al. (2005). 'Closed-loop cortical control of direction using support vector machines'. Neural Systems and Rehabilitation Engineering, IEEE Transactions on **1**3(1):72–80.

W. Penfield & K. Welch (1951). 'The supplementary motor area of the cerebral cortex: a clinical and experimental study'. Archives of Neurology and Psychiatry **6**6(3):289.

B. Pesaran, et al. (2000). 'Temporal structure in neuronal activity during working memory in macaque parietal cortex'. Arxiv preprint q-bio/0309034 .

J. Platt et al. (1998). 'Sequential minimal optimization: A fast algorithm for training support vector machines' .

A. Ramón-Cueto, et al. (1998). 'Long-distance axonal regeneration in the transected adult rat spinal cord is promoted by olfactory ensheathing glia transplants'. The Journal of neuroscience .

P. Roland, et al. (1980). 'Supplementary motor area and other cortical areas in organization of voluntary movements in man'. Journal of Neurophysiology **4**3(1):118–136.

S. Ryu & K. Shenoy (2009). 'Human cortical prostheses: lost in translation?'. Neurosurgical focus **2**7(1):5.

E. Schmidt (1980). 'Single neuron recording from motor cortex as a possible source of signals for control of external devices'. Annals of biomedical engineering **8**(4):339–349.

S. Scott (2000). 'Population vectors and motor cortex: neural coding or epiphenomenon?'. nature neuroscience **3**:307–307.

J. Semmlow (2009a). Biosignal and medical image processing. CRC press.

J. Semmlow (2009b). Biosignal and medical image processing. CRC press.

M. Serruya, et al. (2002). 'Brain-machine interface: Instant neural control of a movement signal'. Nature **4**16(6877):141–142.

E. Stark & M. Abeles (2007). 'Predicting movement from multiunit activity'. The Journal of neuroscience **2**7(31):8387–8394.

A. Suminski, et al. (2009). 'Exploiting multiple sensory modalities in brain-machine interfaces'. Neural Networks **2**2(9):1224–1234.

D. Taylor, et al. (2002). 'Direct cortical control of 3D neuroprosthetic devices'. Science **2**96(5574):1829.

E. Todorov et al. (2000). 'Direct cortical control of muscle activation in voluntary arm movements: a model'. nature neuroscience **3**:391–398.

A. Vallabhaneni, et al. (2005). 'Brain—Computer Interface'. Neural Engineering pp. 85–121.

S. Waldert, et al. (2009). 'A review on directional information in neural signals for brain-machine interfaces'. Journal of Physiology-Paris **1**03(3-5):244–254.

P. Wallisch (2009). MATLAB for neuroscientists: an introduction to scientific computing in MATLAB. Academic Press.

J. Wazen, et al. (2003). 'Transcranial contralateral cochlear stimulation in unilateral deafness'. Otolaryngology–Head and Neck Surgery **1**29(3):248.

J. Wessberg, et al. (2000). 'Letters to Nature: Real-Time Prediction of Hand Trajectory by Ensembles of Cortical Neurons in Primates'. Nature pp. 361–365.

T. Young (1802). 'The Bakerian Lecture: On the theory of light and colours'. Philosophical transactions of the Royal Society of London **9**2:12–48.

Appendix A

A.1 The Cerebral Cortex

Much of interest of modern neuroscience has been given to understand the function and disorders of the brain, and in particular of the human cortex. The human cortex is responsible for much of the planning and execution of action in every day life, such as thinking, perceiving, moving producing and understanding language. Phylogenetically, the cortex is larger in higher primates, in particular humans have the most elaborated and developed cerebral cortex in respect to other spices. The brain cortex is considered the most recent structure in the history of brain evolution and it is also the most highly developed part of the human brain. The portion of the cortex that covers the cerebrum is called cerebral cortex. The cerebral cortex, often referred to as gray matter, is the thin out layer of the cerebral hemispheres. It is formed by grooves, deep furrows or fissures (*sulci*) sided by folded budges *gyri*. The precise reason of this convoluted shape is not clear, an hypothesis is that this configuration increments the surface area increasing the number of neurons in the cortex. Moreover, the number of neurons is directly linked with the power of information storing and processing. The thickness of the cortex is around 2-4 mm and is not significantly changing around the brain surface. [Kandel et al. 2000]

A.1.1 Lobes

The cerbral cortex is anatomically divided into four lobes overlying the cranial bones: frontal, parietal, temporal, and occipital. Each lobes include many distinct functional domain.

Frontal Lobe is involved with decision-making, problem solving and planning

Parietal Lobe is involved in the reception and process of sensor information from the body

Occipital Lobe is involved with vision

Temporal Lobe is involved with memory, emotion, hearing, and language

Two additional regions are the cingulate cortex, which surrounds the dorsal surface of the corpus callosum, and the insula, not visible on the surface. The four lobes are defined by particularly prominent *sulci*, that have a consistent position in the human brain. One of the most prominent indentation is the lateral sulcus or sylvian fessure, that separates the temporal

lobe from the frontal and parietal lobes. Another prominent indentation is the central sulcus, that separates the frontal and parietal lobes.

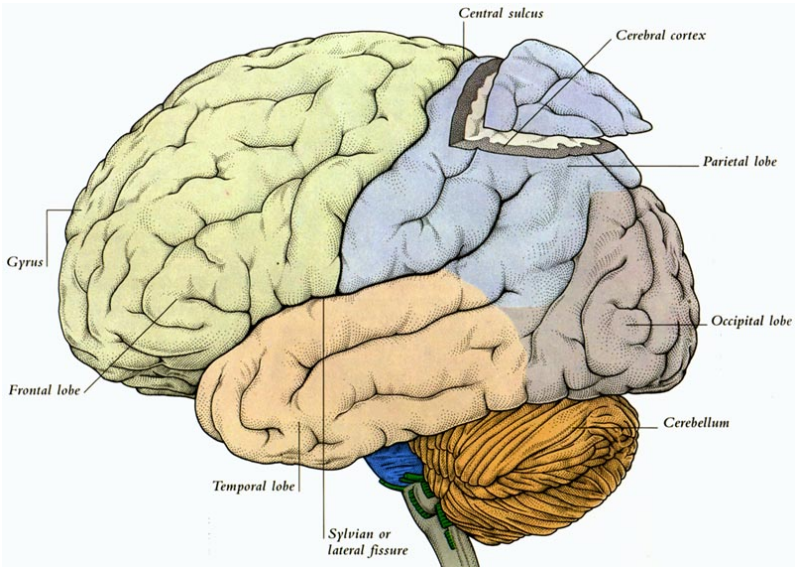

Figure A.1. Division of the brain in four lobes

Many areas of the cerebral cortex are concerned with processing sensory information or delivering motor commands. An area dedicated to a particular sensory modality or motor function includes several specialized area that have different role in processing signals. These areas are known as primary, secondary or tertiary sensory or motor areas depending on the proximity with peripheral sensory or motor pathways. For example the primary motor cortex, located rostral to the central sulcus, is associated with the motor system in the spinal cord. It mediates voluntary movement of limbs and trunk. Cortical cells form the motor cortex influence neurons in the ventral horn of the spinal cord responsible for muscle movements. The primary motor cortex represent the final sites of processing and integration before movement, in fact also other higher-order motors areas located rostraly to the primary motor cortex in the frontal lobe, compute program of movement.

A.1.2 Layers

The noecortex is organized in functional layers, the information is processed across the layers, formed by interconnected set of neurons called *columns*, or *modules*. The number and the function of the layers can vary throughout the cortex. The most typical distribution contains six layers, numbered from the outer surface (pia madre) of the cortex to the white matter. Each layer is characterized by the presence or absence of neuronal cells bodies. Usually layers I to III contains the dendrites of neurons that have the bodies in layers V and VI, and viceversa.

I *Molecular layer* acellular layer. It is occupied by dendrites of cell laying deeper and axon that travel trough or form connection.

II *External granule cell layer* this is the second item external granule cell layer, consternated by small spherical cells.

III *External pyramidal cell layer* which contains a variety of cell types, many with a pyramidal form. Neurons located here are larger than neurons in the most superficial layers.

IV *Internal granule cell layer* which is made by granule (spherical) cells.

V *Internal pyramidal cell layer* which contains pyramidal cells larger than those in layer II.

VI *Polymorphic or multiform layer* which terminates into the white matter, which represents the limit of the cortex.

The distribution and composition of the cortex vary throughout the brain depending on the functional characteristic of the region. Motor cortices, such as the primary motor cortex, have a very meagre layer *IV* but prominent output layer, such as layer *V*. On contrary sensory cortices tend to have a prominent granule layer (IV).

Layer *V* represents also the layer in which the motor information is stored. The neurons belonging to the motor cortex in layer *V* project directly to motor neurons, or interneurons, in the ventral horn of the spinal cord via the corticospinal tract.

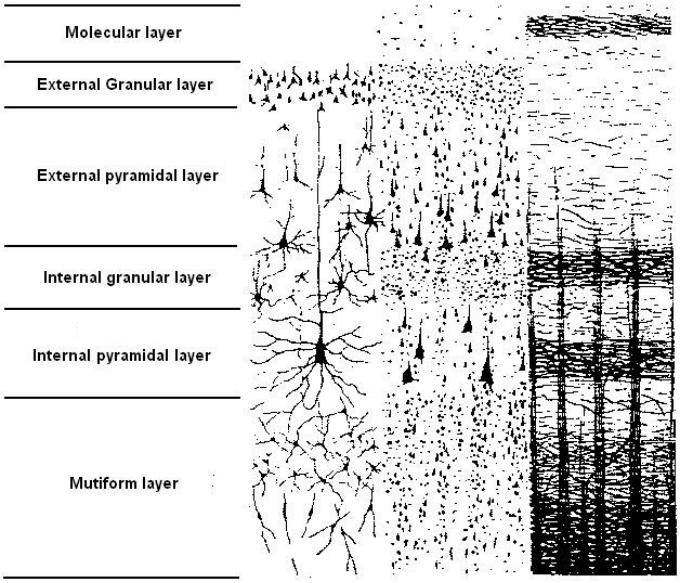

Figure A.2. Stratification of neurons in the brain. [Cyr]

Appendix B

B.1 Artificial neural network

An artificial neural network (ANN), usually called Neural Network (NN), is a mathematical model for classification that is inspired by the structure and/or functional aspects of biological neural networks. A neural network consists of an interconnected group of artificial neurons, and it processes information. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase, that can generate very complex decision boundaries. Modern neural networks are non-linear statistical data modelling tools. They are used to model complex relationships between input and output or to find patterns in data Donna L. Hudson [1992]. A typical three layer neural network is shown in the figure. The bottom layers receives the input and is called the *input layer*, the upper layer of neurons provides the output and is called the *output layer*, the layer in the middle is called the *hidden layer* cause its outputs are not available signals.

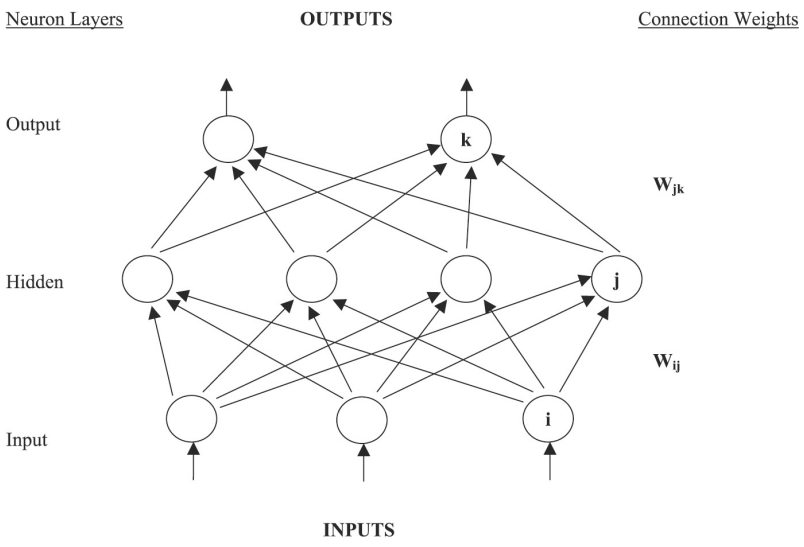

Figure B.1. The basic structure of a three layers artificial neural network.

B.2 The neuron

The motivating factor of calling this mathematical model neural network resides in its structural similarities with the biological nervous system, or biological neural network. In a very generic comparison, from a biological point of view, the neuron is an extension of a simple cell formed by two main appendages: multiple dendrites and the axon. The dendrites have the function of receiving inputs from other neurons and integrate this information. Further they process the information to produce a single output, transmitted by the axon. The cell membrane is characterized by an electrical resting potential of -70 mV. This is maintained by pumping positive ions out of the cell. On the contrary with the other cells a neuron is excitable, depending on the input. If the cell is excited beyond a certain threshold it becomes unable to maintain the resting potential and generate an action potential, that propagates along the axon. The action potential causes the release of neurotransmitter and the message is transmitted to other dendrites.

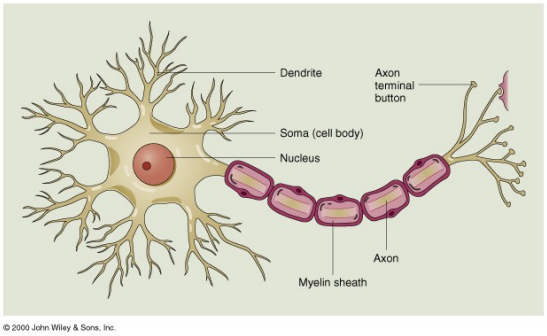

Figure B.2. The basic structure of a neuron.

From an artificial point of view a neuron is the smallest information processing unit in a neural network. An artificial neuron on contrary is characterized by :

Weights factors multiplied with inputs before summing and the ones adjusted and adapted during training

Summation unit unit that sums the products between inputs and weights

Activation function function that transforms the inputs in the neuron outputs

A clear correspondence can be seen between the artificial and the biological neurons: inputs and outputs are dendrites and axons, the weights can be associated with the role of neurotransmitters, the summation unit is like a potential difference and the activation function is working like a neuron firing threshold. Donna L. Hudson [1992]

B.3 Models of the neuron

The first and simplest model of a neuron like a mathematical element able to implement logical operation was presented by McCulloch and Pitts (1943). This elementary unit can have two

binary inputs (0 or 1), a threshold, and a binary output. Such a neuron can implement logical operation like AND and OR. Later, Frank Rosenblatt (1958) used the neuron model to create the first preceprton, a McCulloch and Pitts neuron with an arbitrary number of weighted inputs.

$$a = \sum_{i=1}^{N} w_i x_i + b$$

where x is the input signal, w is the weight, i is the input channel, and N the total number of inputs.

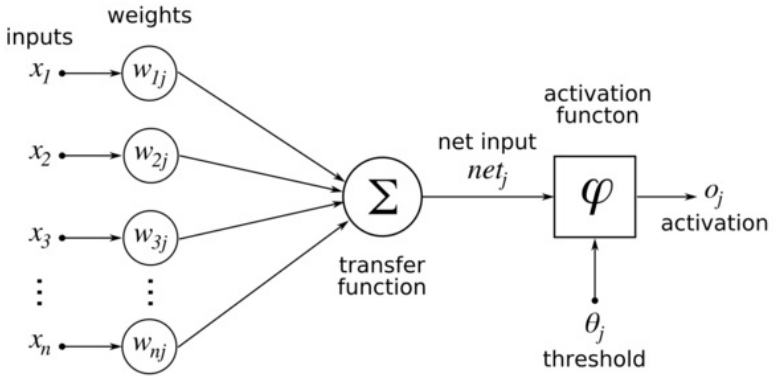

Figure B.3. The basic structure of an artificial neuron.

This new input could be of different magnitudes, not only binary, but the output was still 0 or 1. Also the weight could be different for different neurons, to allow a more powerful computation. Until 1980s there was a sharp decrease in the interest in neural network due to the fact that a single layer perceptron couldn't solve a rather elementary logical function: XOR. This finding implies that all similar networks (es. linear networks) can solve only linearly separable problems. The activation quantity a was passed to a non linear threshold elements that produced an output that could be +1 or -1. The neuron could also have a constant bias input b.

$$\begin{cases} a \geq 0 & +1 \\ a \leq 0 & -1 \end{cases}$$

Because this neuron model had only two output level it could identify only two classes. Multiple classes could be represented by multiple neurons in parallel. A set of parallel McCullogh-Pitts neurons developed by Rosenblatt (1967) were called the preceptron, because of their ability to sense patterns.

A resurgence of neural network was determined by the creation of non-linear multilayer networks, with hidden units and learning rules to train such as backpropagation network. This kind of network can implement non linear (yet differentiable) transfer function, that can approximate any function. Some additional layers can be add, but it can be shown that three layers network can perform all functions of networks with more layer.

Perceptrons (neuron in ANN composed only by two layers) are good in separating an input space into two parts (the output). To obtain a proper partition the perceptron has to be trained

to adjust weights and bias so it can rotate and shift in order to draw a line that perform a proper partition. If the input features are more than two, a hyperplane partition is performed with one dimension less than the input space. Weights and bias are adjusted according to a learning rule:

1. If the output is correct, the weight vector associated with the neuron is not changed
2. If the output is 0 and should have been 1, in the input vector is added to the weight vector
3. If the output is 1 and should have been 0, the input vector is subtracted from the weight vector

It works by changing the weight vector to point more toward input vectors categorized as 1 and away from vectors categorized as 0. A number of variation has been performed by changing the operator that follows the weighting summation that can be linear, hyperbolic tangent, a sigmoid ecc...Combining a sigmoid function at hidden layer and linear function at the output layer can fit any function. Training is done by adapting the free parameters, weight and bias, of the data set to be similar to a certain target. The basic concept of this classifier is the same of the other, minimizing classification errors. [Wallisch 2009]

B.4 McCuoolgh and Pitts neural networks

The McCullough-Pitts neurons can accurately classify any linearly separable training set, generating a linear boundaries between the data, occurring at $a = 0$. The decision boundary can be found solving the following equation:

$$w_i x_i = -b$$

and solving for x_2

$$x_2 = \frac{-w_1}{w_2} x_1 - \frac{b}{w_2}$$

Which is a straight line having a slope of $-\frac{w_1}{w_2}$ and an intercept of $-\frac{b}{w_2}$. If there are three input channels the boundary will be a plan and so on. This neuron is trained using the *preceptron rule*, first applied, where every output is computed, compared with the target. If the difference is 0, the pattern was correctly classified, otherwise weight are changed to reduce the error. The weight should be modify by the product of the error times the weight's input signal, since a larger signal indicated that the associated weight contributes more heavily to the error.

$$w_i(n+1)0w_i(n)) + ex_i(n)$$

$$b_i(n+1) = b_i(n) + e_i$$

Where the error is $e = d - y$, desired output and actual output. That can be corrected regarding the entire data set adding an α term, useful to scale down. It determines how fast the weights converge to a stable solution

$$w_i(n+1) = w_i(n)) + \alpha e x_i(n)$$

$$b_i(n+1) = b_i(n) + \alpha e_i$$

[Semmlow 2009b]

B.5 Gradient Descendent Methods

This method is similar to the mean-squared error methods, and uses the same approximation developed by **Wideow-Hoff**. It consists in comparing targets and outputs and squares the difference to compute the mean square error.

$$MSE = \frac{\sum (output - target)^2}{N}$$

The goal is to minimize the error by taking the derivative of the MSE with respect to each weight, and following the gradient such that the MSE decreases from epoch to epoch.

$$\frac{\partial e_i}{\partial w_i} \approx e_i x_i$$

Since the MSE is a quadratic function will have only one global minimum (if it is any). In other words we are moving into the minimum of an error surface. The **Widrow Hoff** learning rule is expressed mathematically as :

$$\Delta w_{ij} = \epsilon (t_j - post_j) pre_i$$

Where Δ wij is the weight change between input *i* and neuron *j*, ϵ is the learning rate constant, *tj* is the target on neuron *j*, *postj* is the output of postsynaptic neuron *j* and *prei* is the synaptic input *i*. This technique cannot be applied to the McCoullogh and Pitts neurons, given its discontinuous behaviour. [Semmlow 2009a]

B.6 Backpropagation

The backpropagation technique consists in compute an error signal δ and project this error belonging to the output layer back into preceding layers. When the error signal for each neuron is computed the weight coefficients have to be modified. In formulas, $\frac{\partial f(e)}{\partial e}$ represents derivative of neuron activation function (which weights are modified).

$$w'_{(x1)1} = w_{(x1)1} + \eta \delta_1 \frac{\partial f(e)}{\partial e} x_1$$

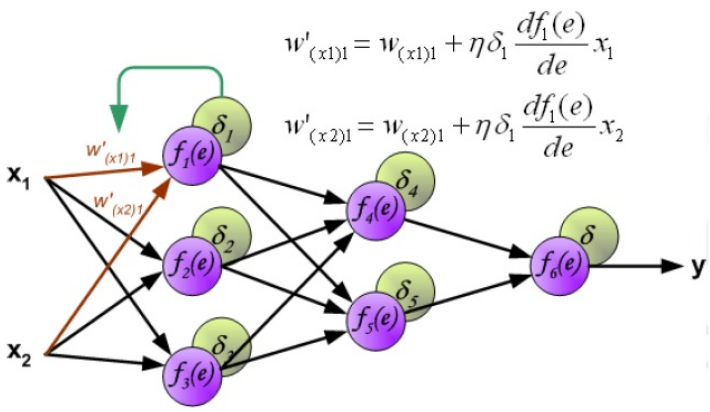

Figure B.4. Example of a beckpropagation step.

Where η affects the network teaching speed. Semmlow [2009a], Duda et al. [2001].

B.7 Different types of neural network

There are different way to implement a neural network. The main types od NN are:

1. Fedforward Neural Network
2. Radial basis function
3. Kohonen self-organizing network
4. Learning Vector Quantization
5. Recurrent Neural Network
6. Modular Neural Network
7. Physical Neural Network

Fedforward NN were the first to be implemented and the most simple. In the feedforward NN the information can propagates only in one direction: forward. The data flow from the input nodes through the hidden nodes and the outputs nodes. No cycle or loops are present in the network. Neurons can be binary (McCulloch-Pitts neurons) or continuous, using sigmoid activation functions in the context of backpropagation error.

Artificial neural network in Matlab

B.8 Initializing Weight

Before starting the training was convenient from a performance point of view to initialize the weights and biases. The network did it automatically, but might be useful initialize it again with the command *init*.

B.9 Feedforward network

Feedforward networks often have one or more hidden layers of sigmoid neurons followed by an output layer of linear neurons. Multiple layers of neurons with nonlinear transfer functions

allow the network to learn nonlinear relationships between input and output vectors. The linear output layer is most often used for function fitting (or nonlinear regression) problems. On the other hand, if you want to constrain the outputs of a network (such as between 0 and 1), then the output layer should use a sigmoid transfer function (such as logsig). This is the case when the network is used for pattern recognition problems (in which a decision is being made by the network). When training large networks, and when training pattern recognition networks, trainscg and trainrp are good choices. Their memory requirements are relatively small, and yet they are much faster than standard gradient descent algorithms [Demuth et al. 1992]. For this reason was chosen a single hidden layer network with a non linear output.

B.10 Training the network

The multilayer feedforward network can be trained both for function approximation or pattern recognition. The training process characterizes a supervised learning, and requires a certain number of good examples represented by data paired with a correct classification. In order to train the network it was necessary to pair the data of the training set with the correct classification. These two elements were the input vector and the target vector. The training purpose was to tun the values of weights and bias in order to optimize the network performance. The default performance function is *mse* the 'mean squared error' between network outputs and the targets. [Demuth et al. 1992]

The two common ways to train a network are the *incremental mod*e and the *batch mode*. In the batch mode all the inputs in the training set are applied to the network before the weight are uploaded. The standard train command applies to the network this modalities. This optimization method use either the gradient of the network performance with respect to the network weights, or the Jacobian of the network errors with respect to the weights. Both of them are calculating using a technique that involves back propagation. It updates the network weights and biases in the direction in which the performance function decreases most rapidly, the negative of the gradient [Demuth et al. 1992]. The default training function in the feedforword network is *trainlm*, using the algorithm of Levenberg-Marquardt, though it performs better for fitting than for pattern recognition.

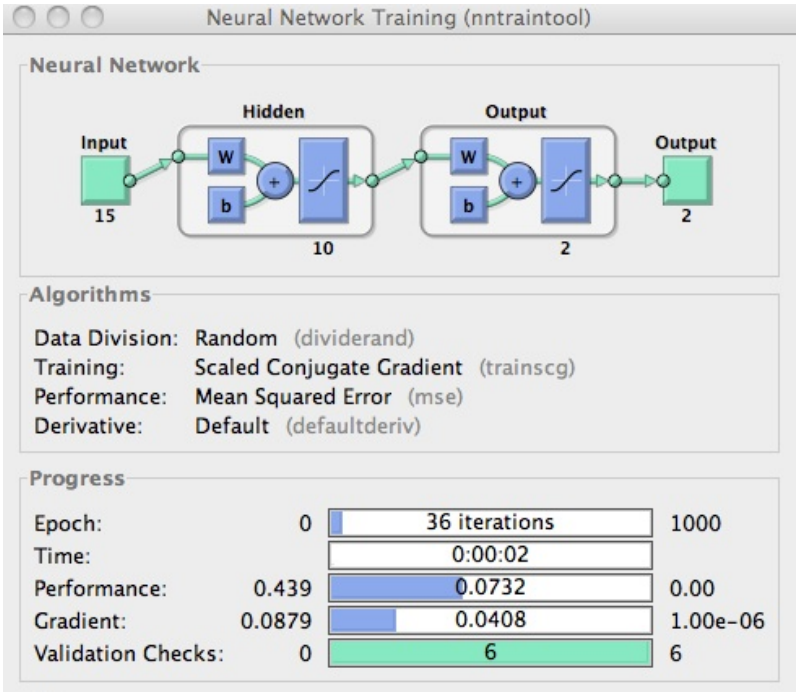

Figure B.5. Train control window

Appendix C

C.1 Support vector machine

Support vector machines (SVMs) are a set of related supervised learning methods used for classification and regression. Sometimes if the training set is not large and it does not reflect very well the characteristic of the test data, linear methods, as the methods based on least mean squares does not perform really well. The problem with the least squares method and number of method working similar is that they compute the decision boundaries based on all the samples in the data set. This give too much importance to points that are not critical to discriminate between classes. Instead sometimes the points that are more critical are the closer to the other classes, as shown in Figure C.1. Support vector machine aim is therefore to produce a better separation of the test data giving more importance to samples that better discriminate between classes. In particular the points closest to the boundary are called *support vectors*, in black. A support vector classifier determines the boundaries that maximizes the distance between support vectors, called margin.

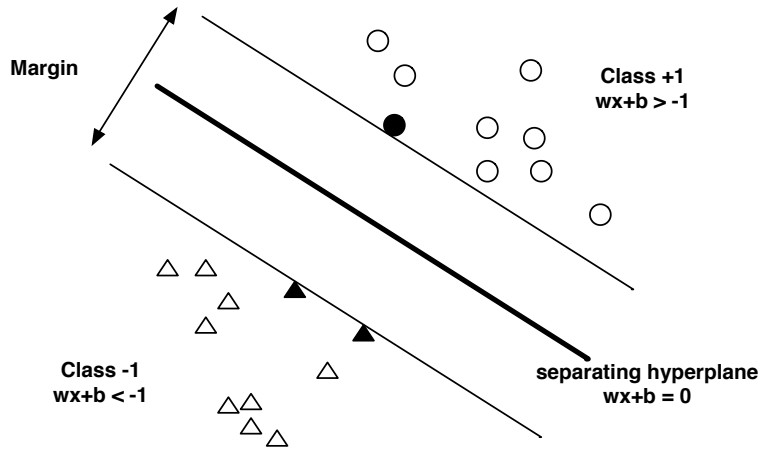

Figure C.1. The points closest to the other classes are more critical in separating the classes and are called support vectors.

In the SVM the classes are assumed to be identified as $\pm$ 1, and the decision boundary is estimate as *y=0*. So using the equation:

$$y = \sum_{i=1}^{N} w_i x_i + b = x_i w + b \tag{C.1}$$

where x_i is the input patterns, w is the weight vector, b the offset. Since the classes are defined as ± 1 the equation for the line dividing the classes will be:

$$x_i w + b \geq 1 \;\; when \;\; y = +1 \tag{C.2}$$

$$x_i w + b \leq 1 \;\; when \;\; y = -1 \tag{C.3}$$

The distance from the hyperplane $x_i w + b = 0$ to the origin is $\frac{-b}{\|w\|}$, where $\|w\|$ is the norm of w, in matrix notation $\sqrt{w^T w}$. The distance from the hyperplane to the origin is:

$$d_0 = \frac{1-b}{\|w\|} \tag{C.4}$$

$$M = \frac{(1-b)}{\|w\|} = \frac{2}{\|w\|} \tag{C.5}$$

where M is the margin. So the maximum margin is obtained by minimizing $\|w\|$. If the data are not linearly separable, some points overlap, in the optimization process M is still to minimize but the constraint is relaxed and some points can be in the other side. The technique described produces linear boundaries and is called linear support vector machine (LSVM), effective when the data are linearly separable. If they are not the error can be accepted or the data can be transformed into higher dimensions [Semmlow 2009a]. In a more general SVM non-linear boundaries can be obtained. Non-linear classifiers can be obtained by applying the kernel trick to maximum-margin hyperplanes. The resulting algorithm is formally similar, except that every dot product is replaced by a non-linear kernel function.

$$F(x) = \sum_{i=1}^{N} w_i k(z_i, x) + b \tag{C.6}$$

The kernel trick is very interesting since provides a bridge from linearity to non-linearity for all the algorithms that depend on dot product between vectors. The kernel allows to map the input data, that behave non-linearly in the input space, into higher dimensional space, where a linear algorithm can operate. Thus though the classifier is a hyperplane in the high-dimensional feature space, it may be non-linear in the original input space. The trick consists in replacing the dot product with a kernel function. For this reason a transformation is done without explicitly mapping the inputs into the new space, that sometimes leads to infeasible solving problem. Various kernel can be used to transform the input space into a features space that is symmetric.

Linear Kernel The Linear kernel is the simplest kernel function. It is given by the common inner product <*x,y*> plus an optional constant *c*. Kernel algorithms using a linear kernel are often equivalent to their non-kernel counterparts, i.e. KPCA with linear kernel is equivalent to standard PCA.

$$k(x, y) = x^T y + c$$

Polynomial Kernel The Polynomial kernel is a non-stationary kernel. It is well suited for problems where all data is normalized.

$$k(x, y) = (ax^T y + c)^d$$

Gaussian Kernel The Gaussian kernel is by far one of the most versatile kernels. It is a radial basis function kernel, and is the preferred kernel when we do not know much about the data we are trying to model.

$$k(x, y) = exp(-\frac{\|x - y\|^2}{2\sigma^2})$$

Transforming the space to higher dimensions can thus result in overfitting. [Platt et al. 1998]and [Lewis 2004]

Appendix D

D.1 Additional tables and results

Rat name	Session	MinWtDe	Min De	Max	Mean	Std
Rat 1	Rat1_08122010	0,38	0,14	0,50	0,33	0,07
Rat 1	Rat1_10122010	0,40	0,25	0,55	0,39	0,07
Rat 1	Rat1_13122010	0,38	0,27	0,56	0,41	0,06
Rat 1	Rat1_14122010	0,32	0,20	0,48	0,36	0,06
Rat 1	Rat1_21122010	0,31	0,22	0,68	0,37	0,07
Rat 1	Rat1_23122010	0,20	0,14	0,52	0,32	0,08
Rat 2	Rat2_29112010	0,46	0,23	0,60	0,41	0,06
Rat 2	Rat2_30112010	0,52	0,23	0,57	0,43	0,07
Rat 2	Rat2_06122010	0,32	0,26	0,48	0,36	0,05
Rat 2	Rat2_07122010	0,38	0,27	0,53	0,38	0,05
Rat 2	Rat2_10122010	0,34	0,27	0,54	0,41	0,06
Rat 2	Rat2_22122010	0,38	0,25	0,63	0,37	0,08
Rat 3	Rat3_15122010	0,41	0,36	0,53	0,44	0,04
Rat 3	Rat3_17122010	0,47	0,38	0,56	0,47	0,04
Rat 3	Rat3_20122010	0,41	0,36	0,58	0,48	0,04
Rat 3	Rat3_21122010	0,36	0,32	0,50	0,39	0,03
Rat 3	Rat3_22122010	0,37	0,26	0,49	0,37	0,05
Rat 3	Rat3_23122010	0,42	0,29	0,49	0,41	0,04
Rat 3	Rat3_07012011	0,43	0,24	0,48	0,34	0,06
Rat 4	Rat4_15122010	0,34	0,22	0,48	0,37	0,06
Rat 4	Rat4_16122010	0,31	0,26	0,48	0,36	0,05
Rat 4	Rat4_17122010	0,31	0,28	0,52	0,39	0,05
Rat 4	Rat4_20122010	0,34	0,22	0,50	0,39	0,05
Rat 4	Rat4_21122010	0,30	0,31	0,52	0,40	0,05
Rat 4	Rat4_23122010	0,36	0,31	0,53	0,41	0,04
Rat 4	Rat4_12012011	0,38	0,34	0,53	0,44	0,04

Figure D.1. **Results for the ANN classifier**. Each value was computed as the mean of the misclassification error rate out of 5 loops. The first three columns represent: rat name, the name of the session and the minimum misclassification error rate for the not denoised data. The others columns refers to denoised data. They represent respectively the minimum, the max, the mean and the standard deviation of the misclassification error rate (out of 100 denoising possibilities).

	Wavelet name									
Th	*db2*	*db4*	*db6*	*coif2*	*coif4*	*coif6*	*sym2*	*sym4*	*sym6*	*haar*
0,4	0	0	0	0	0	0	1	0	1	1
0,5	0	0	0	0	1	0	1	0	0	0
0,6	0	0	0	1	1	1	1	0	0	1
0,8	0	0	0	0	0	0	0	0	2	0
1,0	0	2	0	0	1	0	0	0	1	1
1,2	0	0	0	0	0	0	0	1	0	0
1,4	0	0	1	1	1	0	0	0	0	0
1,6	0	0	0	0	0	0	0	0	0	0
1,8	0	0	2	1	0	1	0	0	2	0
2,0	0	1	1	0	2	0	0	0	0	0

Figure D.2. **Pair of wavelet and threshold for ANN**. The tables summarized the results for all rats. The values correspond to the number of times that a particular pair resulted the best in denoising.

Rat name	Session	Min WtDe	Min De	Max	Mean	Std
Rat 1	Rat1_08122010	0,33	0,19	0,42	0,31	0,05
Rat 1	Rat1_10122010	0,39	0,26	0,48	0,37	0,05
Rat 1	Rat1_13122010	0,54	0,28	0,52	0,41	0,06
Rat 1	Rat1_14122010	0,25	0,22	0,46	0,35	0,05
Rat 1	Rat1_21122010	0,22	0,21	0,49	0,36	0,05
Rat 1	Rat1_23122010	0,25	0,18	0,55	0,35	0,09
Rat 2	Rat2_29112010	0,42	0,24	0,56	0,41	0,06
Rat 2	Rat2_30112010	0,39	0,33	0,57	0,45	0,06
Rat 2	Rat2_06122010	0,46	0,28	0,49	0,40	0,04
Rat 2	Rat2_07122010	0,47	0,29	0,49	0,40	0,04
Rat 2	Rat2_10122010	0,42	0,24	0,51	0,37	0,05
Rat 2	Rat2_22122010	0,35	0,18	0,53	0,35	0,06
Rat 3	Rat3_15122010	0,39	0,35	0,54	0,44	0,04
Rat 3	Rat3_17122010	0,52	0,39	0,62	0,47	0,04
Rat 3	Rat3_20122010	0,36	0,32	0,50	0,41	0,04
Rat 3	Rat3_21122010	0,42	0,28	0,49	0,36	0,04
Rat 3	Rat3_22122010	0,36	0,28	0,44	0,35	0,04
Rat 3	Rat3_23122010	0,47	0,30	0,53	0,41	0,06
Rat 3	Rat3_07012011	0,43	0,22	0,48	0,32	0,07
Rat 4	Rat4_15122010	0,44	0,28	0,52	0,36	0,05
Rat 4	Rat4_16122010	0,40	0,26	0,48	0,37	0,04
Rat 4	Rat4_17122010	0,40	0,22	0,57	0,37	0,05
Rat 4	Rat4_20122010	0,30	0,22	0,50	0,30	0,05
Rat 4	Rat4_21122010	0,31	0,29	0,51	0,37	0,04
Rat 4	Rat4_23122010	0,44	0,30	0,52	0,41	0,04
Rat 4	Rat4_12012011	0,42	0,31	0,57	0,41	0,04

Figure D.3. **Results for the SVM classifier**. Each value was computed as mean of the misclassification error rate out of 5 loops. The first three columns represent: rat name, the name of the session and the minimum misclassification error rate for the not denoised data. The others columns refers to denoised data. They represent respectively the minimum, the max, the mean and the standard deviation of the misclassification error rate (out of 100 denoising possibilities).

	Wavelet name									
Th	*db2*	*db4*	*db6*	*coif2*	*coif4*	*coif6*	*sym2*	*sym4*	*sym6*	*haar*
0,4	1	0	0	0	0	1	1	0	0	0
0,5	2	0	0	0	1	0	2	0	1	0
0,6	1	1	0	1	0	0	1	1	2	0
0,8	0	0	0	2	0	0	0	1	0	0
1,0	1	0	1	0	0	0	1	0	1	1
1,2	1	1	1	1	1	0	1	0	1	0
1,4	0	0	0	0	0	0	0	1	0	0
1,6	0	0	0	0	0	0	0	0	0	1
1,8	0	1	0	0	0	0	0	0	1	0
2,0	0	0	0	0	0	0	0	0	0	0

Figure D.4. **Pair of wavelet and threshold for SVM**. The tables summarized the results for all rats. Each value was computed as mean of the misclassification error rate out of 5 loops. The values correspond to the number of times that a particular pair resulted the best in denoising.

Rat name	Session	Min WtDe	Min De	Max	Mean	Std
Rat 1	Rat1_08122010	0,20	0,00	0,43	0,17	0,09
Rat 1	Rat1_10122010	0,19	0,06	0,50	0,24	0,10
Rat 1	Rat1_13122010	0,25	0,10	0,45	0,28	0,07
Rat 1	Rat1_14122010	0,18	0,09	0,41	0,24	0,06
Rat 1	Rat1_21122010	0,17	0,00	0,33	0,15	0,12
Rat 1	Rat1_23122010	0,00	0,00	0,33	0,06	0,09
Rat 2	Rat2_29112010	0,28	0,06	0,50	0,27	0,09
Rat 2	Rat2_30112010	0,50	0,09	0,50	0,30	0,10
Rat 2	Rat2_06122010	0,20	0,13	0,43	0,25	0,06
Rat 2	Rat2_07122010	0,25	0,13	0,50	0,27	0,06
Rat 2	Rat2_10122010	0,29	0,07	0,50	0,29	0,09
Rat 2	Rat2_22122010	0,13	0,00	0,50	0,22	0,10
Rat 3	Rat3_15122010	0,33	0,26	0,50	0,35	0,05
Rat 3	Rat3_17122010	0,38	0,26	0,53	0,40	0,05
Rat 3	Rat3_20122010	0,36	0,14	0,50	0,37	0,08
Rat 3	Rat3_21122010	0,31	0,22	0,41	0,30	0,04
Rat 3	Rat3_22122010	0,29	0,11	0,41	0,25	0,05
Rat 3	Rat3_23122010	0,24	0,15	0,44	0,29	0,06
Rat 3	Rat3_07012011	0,35	0,09	0,44	0,24	0,06
Rat 4	Rat4_15122010	0,29	0,04	0,43	0,27	0,07
Rat 4	Rat4_16122010	0,27	0,12	0,42	0,25	0,06
Rat 4	Rat4_17122010	0,25	0,05	0,45	0,26	0,08
Rat 4	Rat4_20122010	0,19	0,06	0,44	0,25	0,08
Rat 4	Rat4_21122010	0,19	0,12	0,46	0,29	0,07
Rat 4	Rat4_23122010	0,27	0,20	0,47	0,33	0,05
Rat 4	Rat4_12012011	0,29	0,18	0,50	0,34	0,06

Figure D.5. **Results for the ANN classifier**. Each value was computed as minimum of the misclassification error rate out of 5 loops. The first three columns represent: rat name, the name of the session and the minimum misclassification error rate for the not denoised data. The others columns refers to denoised data. They represent respectively the minimum, the max, the mean and the standard deviation of the misclassification error rate (out of 100 denoising possibilities).

	Wavelet name									
Th	*db2*	*db4*	*db6*	*coif2*	*coif4*	*coif6*	*sym2*	*sym4*	*sym6*	*haar*
0,4	1	2	3	2	1	2	3	1	2	1
0,5	2	0	2	2	1	2	2	2	2	1
0,6	1	2	1	4	3	2	3	2	0	1
0,8	1	0	2	2	2	1	2	1	1	3
1,0	1	2	2	1	1	1	0	2	3	0
1,2	1	1	3	2	1	2	1	3	2	2
1,4	0	1	1	2	2	2	1	1	1	2
1,6	2	2	1	1	1	1	3	1	0	1
1,8	2	0	1	1	2	1	0	2	2	0
2,0	1	1	0	1	2	0	1	0	0	1

Figure D.6. **Pair of wavelet and threshold for ANN**. The tables summarized the results for all rats. Each value was computed as min of the misclassification error rate out of 5 loops. The values correspond to the number of times that a particular pair resulted the best in denoising.

Rat	Session	Min WtDe	Min De	Max	Mean	Std
Rat 1	Rat1_08122010	0,20	0,00	0,30	0,14	0,08
Rat 1	Rat1_10122010	0,31	0,06	0,31	0,20	0,07
Rat 1	Rat1_13122010	0,45	0,15	0,45	0,30	0,07
Rat 1	Rat1_14122010	0,14	0,09	0,41	0,24	0,06
Rat 1	Rat1_21122010	0,17	0,00	0,33	0,12	0,11
Rat 1	Rat1_23122010	0,00	0,00	0,36	0,14	0,12
Rat 2	Rat2_29112010	0,25	0,13	0,44	0,29	0,07
Rat 2	Rat2_30112010	0,30	0,14	0,50	0,32	0,08
Rat 2	Rat2_06122010	0,37	0,20	0,47	0,31	0,06
Rat 2	Rat2_07122010	0,42	0,13	0,42	0,29	0,06
Rat 2	Rat2_10122010	0,22	0,07	0,43	0,24	0,08
Rat 2	Rat2_22122010	0,25	0,00	0,38	0,16	0,09
Rat 3	Rat3_15122010	0,32	0,21	0,50	0,35	0,05
Rat 3	Rat3_17122010	0,40	0,28	0,50	0,38	0,05
Rat 3	Rat3_20122010	0,26	0,11	0,41	0,31	0,06
Rat 3	Rat3_21122010	0,31	0,13	0,47	0,29	0,06
Rat 3	Rat3_22122010	0,22	0,12	0,36	0,25	0,06
Rat 3	Rat3_23122010	0,41	0,15	0,50	0,32	0,06
Rat 3	Rat3_07012011	0,39	0,09	0,44	0,24	0,07
Rat 4	Rat4_15122010	0,39	0,07	0,43	0,26	0,07
Rat 4	Rat4_16122010	0,35	0,12	0,38	0,28	0,06
Rat 4	Rat4_17122010	0,30	0,10	0,50	0,26	0,07
Rat 4	Rat4_20122010	0,18	0,00	0,33	0,19	0,07
Rat 4	Rat4_21122010	0,23	0,14	0,42	0,27	0,06
Rat 4	Rat4_23122010	0,35	0,11	0,47	0,31	0,06
Rat 4	Rat4_12012011	0,32	0,11	0,50	0,31	0,07

Figure D.7. **Results for the SVM classifier**. Each value was computed as minimum of the misclassification error rate out of 5 loops. The first three columns represent: rat name, the name of the session and the minimum misclassification error rate for the not denoised data. The others columns refers to denoised data. They represent respectively the minimum, the max, the mean and the standard deviation of the misclassification error rate (out of 100 denoising possibilities).

	Wavelet name									
Th	*db2*	*db4*	*db6*	*coif2*	*coif4*	*coif6*	*sym2*	*sym4*	*sym6*	*haar*
0,4	1	2	4	0	2	3	1	1	1	2
0,5	1	2	3	2	0	0	1	0	1	2
0,6	1	1	1	5	1	2	1	2	4	1
0,8	2	2	1	1	1	0	2	1	1	1
1,0	1	1	2	2	1	1	1	0	2	2
1,2	1	1	1	4	1	3	1	3	2	1
1,4	0	1	1	2	4	2	0	2	2	2
1,6	1	1	0	2	3	2	1	2	1	1
1,8	0	1	1	2	4	2	0	2	1	1
2,0	0	1	1	2	3	2	0	2	0	0

Figure D.8. **Pair of wavelet and threshold for SVM**. The tables summarized the results for all rats. Each value was computed as min of the misclassification error rate out of 5 loops. The values correspond to the number of times that a particular pair resulted the best in denoising.

Rat name	Session	Min ANN	Min SVM
Rat 1	Rat1_08122010	0,14	0,19
Rat 1	Rat1_10122010	0,25	0,26
Rat 1	Rat1_13122010	0,27	0,28
Rat 1	Rat1_14122010	0,20	0,22
Rat 1	Rat1_21122010	0,22	0,21
Rat 1	Rat1_23122010	0,14	0,18
Rat 2	Rat2_29112010	0,23	0,24
Rat 2	Rat2_30112010	0,23	0,33
Rat 2	Rat2_06122010	0,26	0,28
Rat 2	Rat2_07122010	0,27	0,29
Rat 2	Rat2_10122010	0,27	0,24
Rat 2	Rat2_22122010	0,25	0,18
Rat 3	Rat3_15122010	0,36	0,35
Rat 3	Rat3_17122010	0,38	0,39
Rat 3	Rat3_20122010	0,36	0,32
Rat 3	Rat3_21122010	0,32	0,28
Rat 3	Rat3_22122010	0,26	0,28
Rat 3	Rat3_23122010	0,29	0,30
Rat 3	Rat3_07012011	0,24	0,22
Rat 4	Rat4_15122010	0,22	0,28
Rat 4	Rat4_16122010	0,26	0,26
Rat 4	Rat4_17122010	0,28	0,22
Rat 4	Rat4_20122010	0,22	0,22
Rat 4	Rat4_21122010	0,31	0,29
Rat 4	Rat4_23122010	0,31	0,30
Rat 4	Rat4_12012011	0,34	0,31

Figure D.9. Comparison between minimum misclassification error rate obtained for the ANN and the SVM with denoising algorithms. Each value was computed as mean of the misclassification error rate out of 5 loops.