INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Julia Sanchez
julia.sanchez@insa-lyon.fr
Génie électrique
Roeder S.R.L

Internship report

Implementation of an automatic arrythmias classification with neural networks

Tuteur : Philippe Delachartre

from 01/04/2016 to 29/08/2016

Abstract

This report synthesizes the work realized at the enterprise ROEDER S.R.L, and what I learned during my Master's Internship. This work concludes my formation at the National Institute of Applied sciences of Lyon (INSA Lyon) in Electrical Engineering. It is directly linked to the option I chose, which is Signal and Image processing and the master "Systèmes et images" I realized. The goal of this project was to classify automatically different arrhythmia types. The methods presented here are based on neural networks and the MIT-BIH data base was used. The project can be divided in three main parts. Firstly, the signals features were extracted. Secondly, the nets were implemented and thirdly, the methods were evaluated. At the end, the best method was selected and implemented in C in order to include the program to the equipment.

Contents

1 Project background

1.1 Enterprise description

ROEDER S.R.L is a firm specialized in business of biomedical products in various fields (cardiology, urology, anestesy y otorrinolaringology). It offers products of the worldwide leading firms. Roeder S.R.L. was created in 1993, selling products from RÜSCH and Pilling Weck enterpises. Since then, the organization has set its place in the market as provider of first quality products, and supplies clinics, and hospital in all latin America. In the field of electrophysiology, ROEDER dealt with equipments of the well known firms Cardioline, CardioTek, DM Software, Fieger-Endotech, Osypka and Scmitz.

Three years ago, the enterprise included to its mission the production in Argentina of cardiological equipments and softwares. It also carries out the technic assistance and a user formation for the clients. ROEDER S.R.L. hired an ingeneering team to form the department of research and development in order to develop the most innovative electrocardiograph used in electrophysiological studies. The equipment is called "Cardian". It is modular, adjustable and portable and can be used in the framework of control examinations for no specialized doctors or for physiological studies. These studies are handled with electrodes located on the heart through the use of catheters. It allows the doctor to burn the nerves which are not working in an accurate way by radiofrequency.

The tutor for this internship was Marcelo Lerendegui who is the responsable of the ingeneering team.

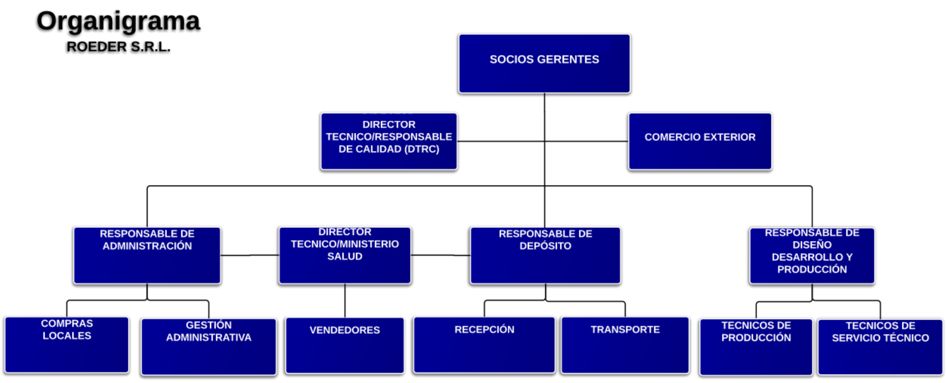


Figure 1: Organigrama de la empresa Roeder S.R.L.

1.2 Project overview and background

1.2.1 ECG signals

A study of the ECG signal was done. The signal is composed of three waves. The first is the P wave. It is due to the depolarization of the atriums. the second is the QRS complex, it is the consequence of the depolarization of the ventricles. The last one is the T wave and it corresponds to the repolarization of the ventricles. A study of the arrhythmia detection by doctors was realized. The meditions are taken by means of electrodes placed on the patient's body. Commonly, from 6 to 12 electrodes are used. A part of the electrodes are located in the wrists and the ankles, they are refeared as D. The lead DII is the addition of DI and DIII. The other part is placed directly on the chest in order to get all signals passing through the heart, they are refeared as V. The best leads to oberv the beats with all features are DII and V1. From the time intervals between the signals, doctors can deduce the angle of the heart which is an important feature to make a diagnosis. According to [1], the abnormalities are better seen on the lead V1. But the normal beats are better displayed on the DII. This study revealed that the best features to focus on are :

- The width of each wave
- An aberrated form of a wave
- The absence of a wave
- The overlap of waves
- The time interval between waves
- The heart frequency in the surroundings of the beat

A particular care must be taken on the age of the patient because children don't present the same structure of the ECG signal as adults. They must be treated separately.

1.2.2 State of the art

In the last ten years, the development of the neural networks made appear accurate softwares in cardiologic equipments like *holders*. These equipments are used for long studies (from 30 minutes to some days) to detect anomalies. In order to make the doctors'work easier, a previous automatic problems detection is done. However, nowadays, this kind of automatic detection can't be trusted and needs to be inspected by specialists. This project aimed at developping a program accurate enough to implement it to an electrocardiograph in order to be used by doctors no specialists (emergency doctors for instance). The focus needed to be set on the rejection of false negatives diagnosis. In order to realize this project, a study of the state of the art was done. In all cases, the electrocardiographic signal is divided in beats and each of them is processed and entered to a neural network. This study highlighted two main investigation areas for the project. One is to select the good neural network, the other is to select the features of the beat studied that best characterize the class of arrhythmia it belongs to.

Three kinds of beat features exist : morphological ones, dynamical ones (as shown in[4]) and information of the patient studied. First ones refear to the datas extracted from one beat. Various methods were implemented. The signal can be decomposed with Hermite coefficients as it is done in[7, 13], with wavelets as in[6, 4], with an autoregressive modelisation as in[9], by Principal Component Analysis as used in[3, 10] or by Independent Component Analysis as in[4]... Second ones refear to the relation between the beat and the complete signal. Indeed, it takes into account the time intervals between beats. The last ones can be varied such as the patient's sexe, age or medical information (diabete, anterior heart attack...).

Numerous neural networks were developed. They can be separated in two classes : the supervised and the unsupervised nets. The difference lies in the datas used for training the net. The first one needs classes to be trained and the second one doesn't need any previous information. The most used in the arrhythmias classification are the supervised nets because the knowledge of the doctors can be leveraged . Firstly, the multilayer perceptron (MLP) is studied in [5]. This is a basic network that consists in entering signals features to a black box and getting the classification at the output. This network is trained entering examples and adapting the net to get the output expected. Various algorithms are used. A list of some of them is given in[9]. Among them, the most used are the gradient descent, the resilient backropagation, the Newton method, the Levenberg-Marquardt algorithm, the scaled conjugate gradient, and the Fletcher-Powell Conjugate Gradient. Secondly, the autoassociative neural network can be used. They have a similar structure to the MLP. However, the goal is to link one beat to itself in the ouput. Indeed, this kind of network is trained with a single arrhythmias class. In other words, the same number of networks and of classes is set. This method aims at adapting the network to get at the output the exact input if the beat belongs to the class studied or a shape very different if it doesn't. This network is useful only for the morphological features of the heart beats.

Thirdly, the Self Organizing Map (SOM) [12] can be used to classify the heart signals. This is an unsupervised neural network. The key is to map the training examples in 2D and to train the net to gather the examples of the same group. Then, the goal is to recognize the areas in which the different classes are stored.

The process can be realized in any derivation of the electrocardiograph. Therefore the issue of the fusion of the datas must be studied. Two main methods are used. The heartbeats that lead to different classifications can be rejected. The probability of certainty can be computed for each lead and the result the most likely is selected. These two methods are used in [4].

To summary, the pists of investigation are the decomposition method of the heartbeats, the other features that can be extracted from the whole signal or patient, the kind of neural network used, the kind of algorithm used to iteratively get closer to the expected outputs and the type of evaluation used to classify the beats. In the next parts, some of the previous methods will be developped and evaluated.

1.3 Data base MIT-BIH

Two data bases were available. The UCI data base and the MIT-BIH data base. The UCI one is composed of features extracted from the beats whereas the MIT-BIH contains the signal without processing. In order to be close to the datas available in the *Cardian* equipment, the MIT-BIH one was selected. It is composed of 48 thirty minutes long ECG records extracted from 47 different patients. These meditions were taken in the Beth

Israel Hospital Arrhythmia Laboratory between 1975 and 1979. 23 records were selected randomly and 25 records that contain special phenomenas were added in order to give an important number of samples for each arrhythmia class. The subjects were 25 men aged 32 to 89 years, and 22 women aged 23 to 89 years. All the heartbeats were annotated by doctors. There exist 41 different annotations that mix arrhythmia types, comments on the signal or remarks on the particular shape of the beat studied. Each signal is given with an header file containing information about the patient and the recording (sampling frequency, derivation, age, sexe). The derivations used can be DII, v1, v2, v4. The leads used in this project were the DII and the v1. The annotations that were kept are :

- Normal beat
- Left bundle branch block beat
- Right bundle branch block beat, aberrated atrial premature beat, atrial premature contraction which are gathered in the Atrial premature beat signals.
- Premature or ectopic supraventricular beat
- Premature ventricular contraction
- Paced beat
- Signal quality change used to reduce the noise.

An equivalence table is displayed figure 1 . These initials will be used in the whole report.

N	LBBB	RBBB	APC	PVC	PC
Normal	Left bundle branch block	Right bundle branch block	Atrial premature beat	Ventricle Premature beat	Paced beat

Table 1: Initials equivalence

2 Project description

2.1 Design brief

Reminder of the enterprise needs : « *Developping a graphic interface able to compare arrhythmias classification methods on all data bases of ECG signals. The methods will need to classify 6 types of arrhythmias. The methods will be based on neural networks. The focus must be put on the minimal rate of false negative classifications. Including the best method to the equipment.* ».

2.1.1 Network

The decomposition of the signals must be independent from the scale or the sampling frequency of the signal. The training part must be quick enough to realize tests. It can be implemented in an equipment in order to improve the training during the use by doctors. The rejection of abnormal beats must be minimized. The evaluation must be realizable in real time. The process can be realized for any class number or derivations number.

2.1.2 Graphic interface

It must be simple to use, and educational. The users must be able to test various signals and to understand the advantages and disadvantages of the different methods implemented.

a) Main axis

The interface must be composed of three parts : the extraction of the beats features, the training of the network, the evaluation of the results.

b) Parameters

The main parameters that must appear are the following:

- Features of the signal decomposition (morphologic coefficients).
- Features of the other coefficients extraction (dynamic and patient's information).
- Type of neural network chosen.
- Parameters of the net (number of neurons per layer, constants for algorithms of diminution of cost function, initialization).
- Type of evaluation/fusion chosen.

Each parameter must be supported by an explication of its characteristics.

c) Interface outputs.

The interface must display the graphics of the evolution of the networks during the training. It must contain the following evaluation tools :

- Confusion matrix
- Specificity
- Sensibility
- Accuracy
- Rate of abnormal beats that are not detected.

It might be inspired by matlab neural network toolbox.

2.2 Tasks overview

The main tasks that were realized are summarized in the Gantt below:

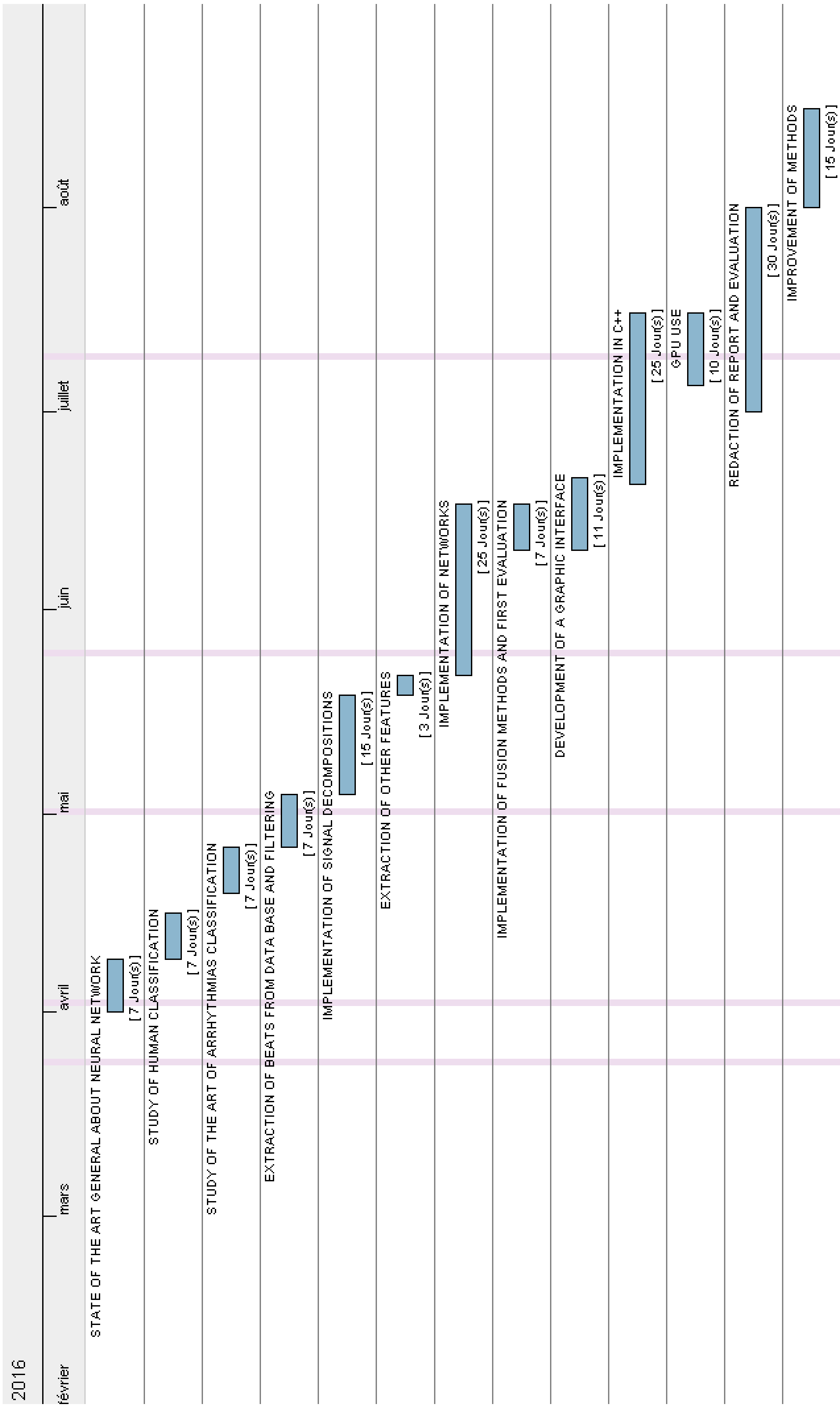


Figure 2: Gantt scheme of tasks

The Gantt scheme planned for this project is represented figure 3. The implementation of the methods was faster than expected. However, the evaluation and the redaction of reports took more time and needed to be divided all along the project. A C-implementation was added because the result were good enough to include the program to the equipment.

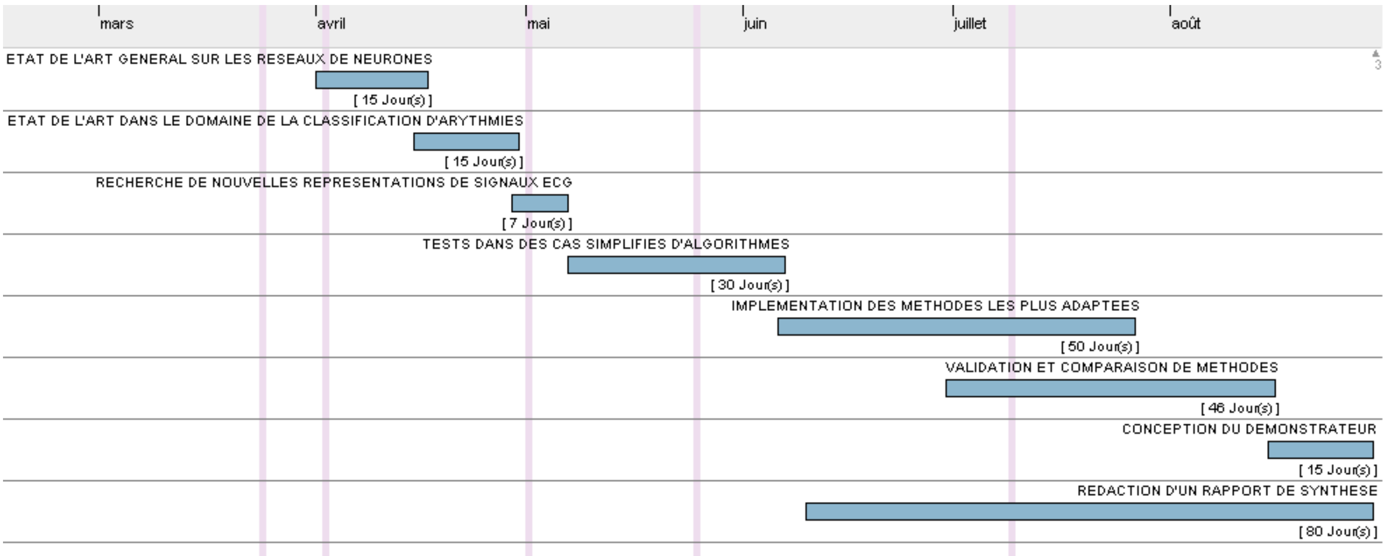


Figure 3: Initial Gant scheme

3 Main algorithms implementation

3.1 Preprocessing

In order to process the signals, some artifacts must be eliminated. An extract of the signal is shown figure 4. The heart signal is contained between 0.5 and 40 Hz[8]. Two kinds of artifacts might appear adquiring the signals, the baseline wander and a high frequency noise.

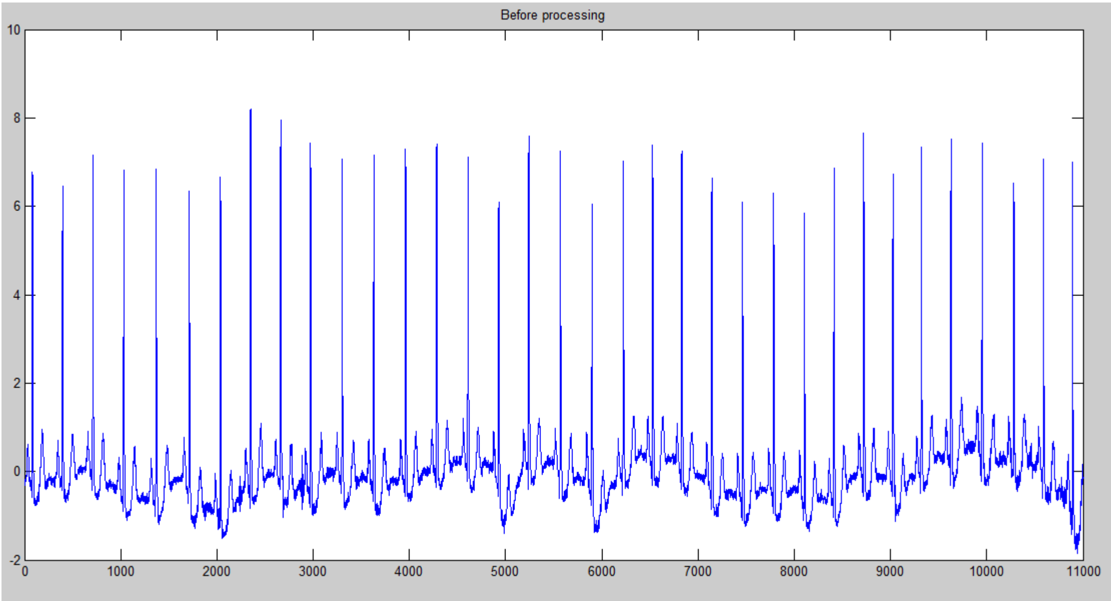


Figure 4: Signal before processing

Firstly, the baseline wander was removed. This noise comes from the movements of the patient and generally corresponds to its respiration. Therefore its frequency is contained between 0.15 and 0.3 Hz. In order to do so, a wavelet decomposition was done to the order 8. At this step, the details were kept and the mean components were rejected.

Secondly, the high frequency noise was removed using an anisotropic filter. This filter was implemented in order to preserve the beats shape. Indeed, it consists in computing, for each point, the mean value of all the points in a neighbourhood. However, a weight is attributed to each neighbour to take into account its distance value with the point studied. This method permits to filter the noise with a weak gradient and to not filter when the gradient is high as for the edge of the heart pulses. In the figures 5 and 6, the difference between the result using band pass commun filters and the result with the filters proposed here is highlighted.

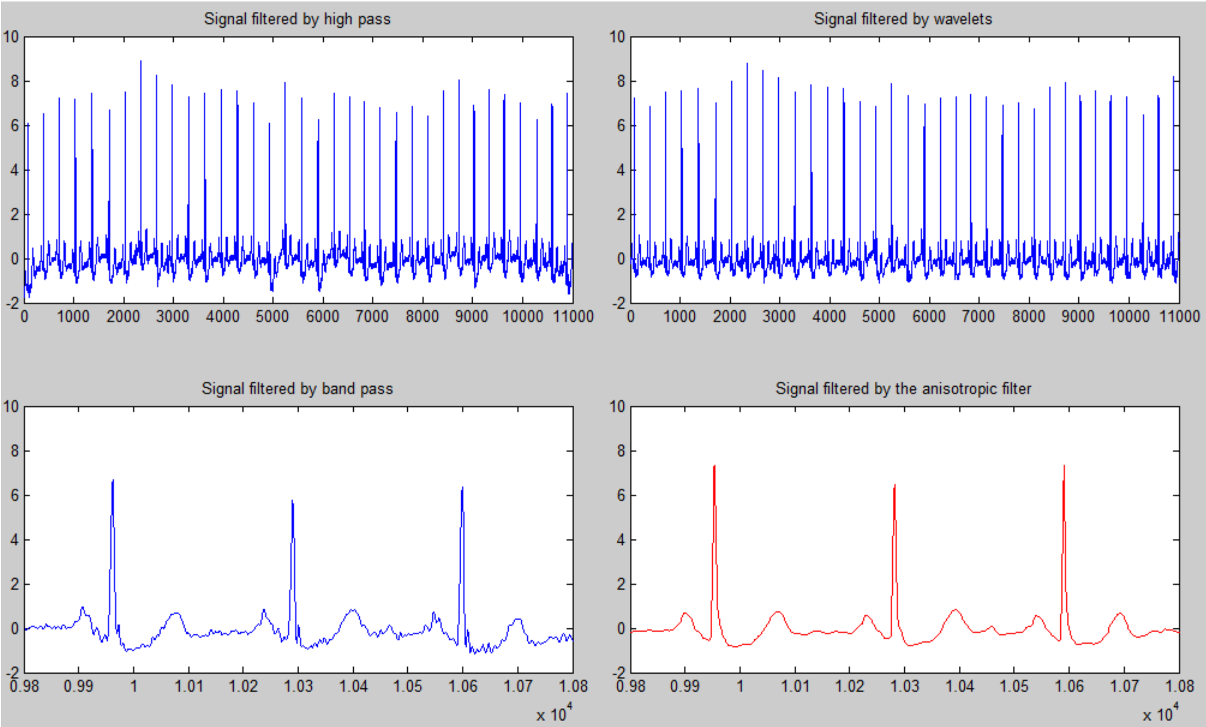


Figure 5: Difference between the commun process and the method proposed

It can be noticed that the baseline wander is better eliminated with the method proposed and the beats are better aligned on the signal at the top right of the figure 5. The result is clearly better filtered and the shape of the beats and the gradient is preserved.

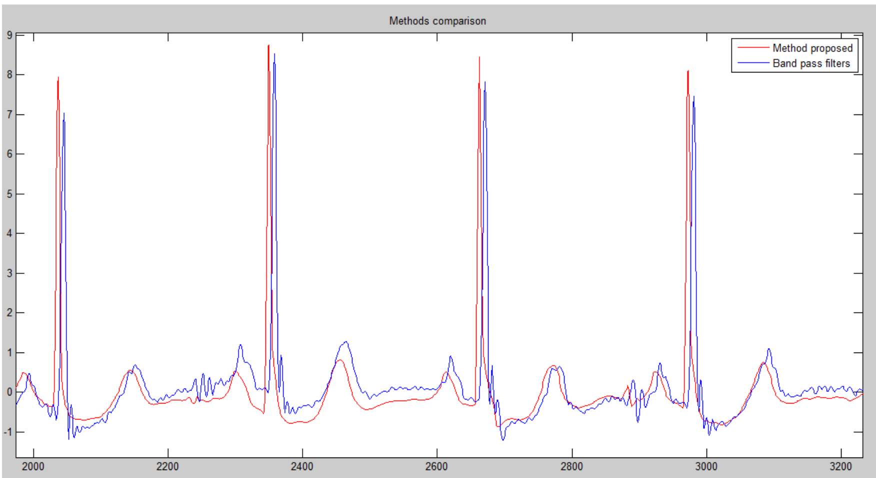


Figure 6: Comparison of results

3.2 Signal decomposition and features extraction

In the project, four signal transformations were implemented. In this report, three of them are described : Hermite transformation, wavelets transformation, PCA.

3.2.1 Hermite

The Hermite polynomials are very useful to transform the signals.

The Hermite functions are built with Hermite polynomials and form an orthonormal basis. The Hermite functions are expressed mathematically in the equation 1.

$$\phi(t) = \frac{1}{\sqrt{\sigma 2^n n! \sqrt{\pi}}} e^{-t^2/2\sigma^2} H_n(t/\sigma) \tag{1}$$

with $\mathrm{H_n}$the Hermitian polynomial order n. The role of the parameter $\sigma$is described below.

The Hermitian polynomials can be expressed recursively as in the equation 2.

$$2xH_{n-1}(x) + 2(n-1)H_{n-2}(x) \tag{2}$$

The six first Hermite polynomials are listed in 3.

$$\begin{gathered} H_0(x) = 1, \\ H_1(x) = 2x, \\ H_2(x) = 4x^2 - 2, \\ H_3(x) = 8x^3 - 12x, \\ H_4(x) = 16x^4 - 48x^2 + 12, \\ H_5(x) = 32x^5 - 160x^3 + 120x \dots \end{gathered} \tag{3}$$

The shape of these functions is adapted to represent the heartbeats as it can be notice figure 7.

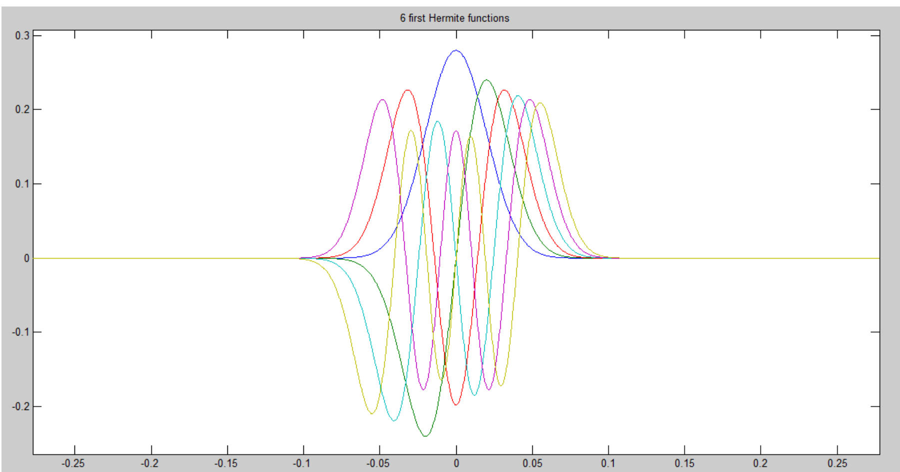
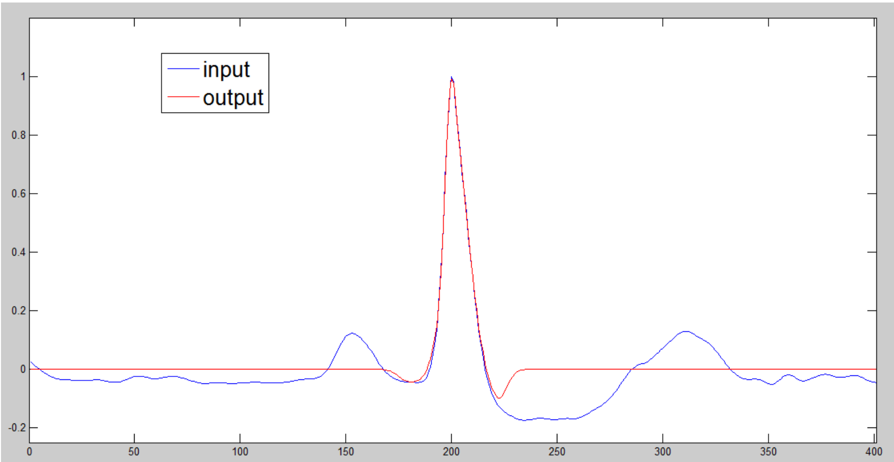
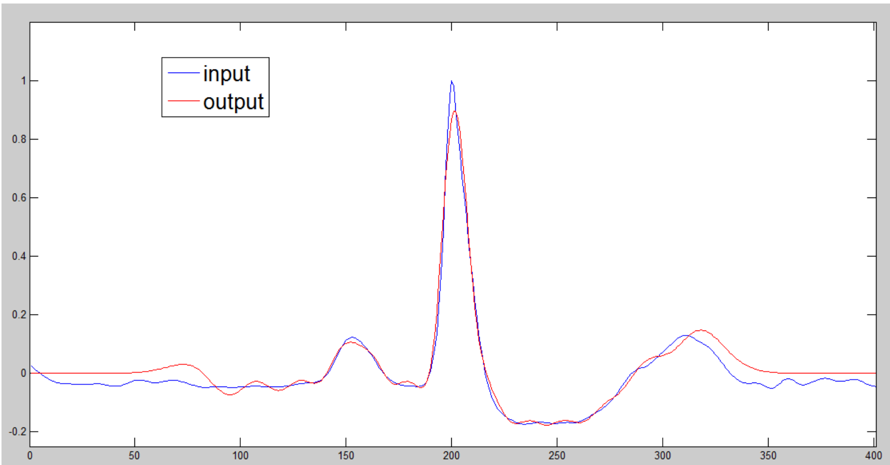


Figure 7: 6 first Hermitian functions

The parameter σ permits to handle the width of the Hermitian functions. Indeed, the Hermite functions must be wide enough to cover the three waves P, R and T. However, the more coefficients are taken, the more the width covered by the Hermite waves increases. Then, the goal is to find a good compromise between the parameter σ and the number of coefficients taken. In the figure 8, it can be notice the effect of the variation of the parameter sigma over the representation of the signal. In the figure a ($\sigma = 0.01$), the width of the representation of the signal is shorter than in figure b ($\sigma = 0.05$) but the accuracy of the R peak is better.



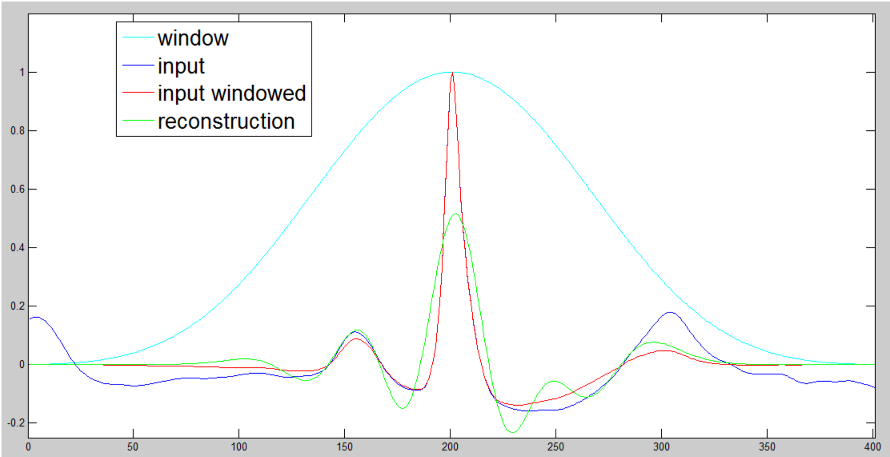
(a) $\sigma = 0.01$



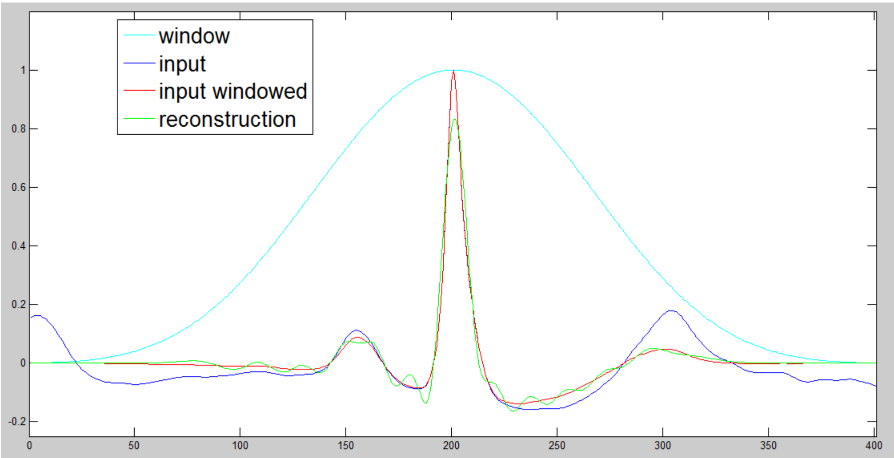
(b) $\sigma = 0.05$

Figure 8: Reconstruction by Hermite transform with 25 coefficients

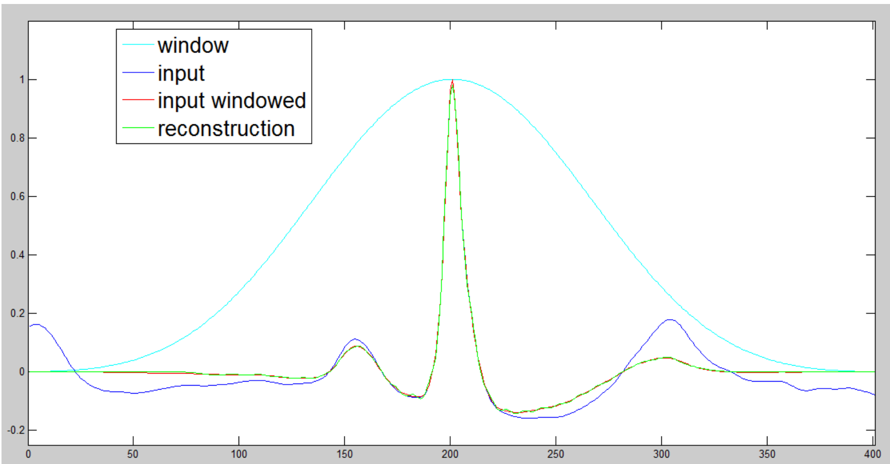
To realize the transform, the heart beat studied is first passed through a window that permits to get a soft edge that leads to 0 in the two sides of the beat. The Kaiser window was chosen. A parameter β permits to change the width of the window. In the figure 9, the steps of the decomposition is detailed. The link between the coefficients numbers and the parameter σ described previously is emphasized.



(a) 10 coefficients $\sigma = 0.09$



(b) 25 coefficients $\sigma = 0.05$



(c) 50 coefficients $\sigma = 0.035$

Figure 9: Reconstruction by Hermite transform

The result is shown figure 10 for 50 coefficients after passing through the inverse of the Kaiser window.

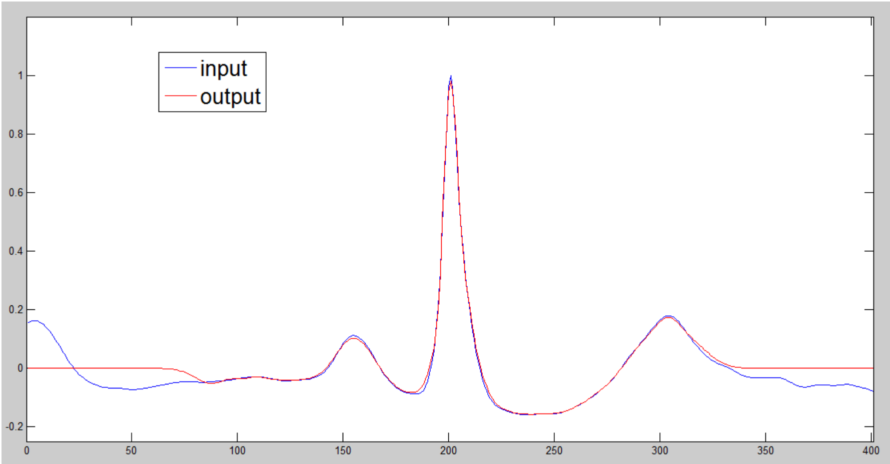


Figure 10: Reconstruction with 50 coefficients

3.2.2 Wavelets

To overcome the issue of width of the base signals, the best way is to use wavelets. The wavelets can be adapted in time to the heartbeat. The Daubechies wavelets can be used to get useful features of the signals. Indeed, the shape of these wavelets is very close to the heartbeat as it can be notice in the figure 12. We chose the order 6 to get the closest possible to the heartbeat shape.

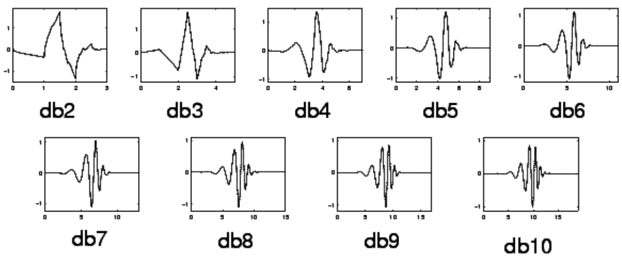


Figure 11: Daubechies wavelets

In the figure 12, it can be noticed that the power of the signal is concentrated in few coefficients. This figure highlights the following coefficients:

$$[5, 14]\ U\ [19, 31]\ U\ [36, 48]\ U\ [70, 75]\ U\ [117, 121] \tag{4}$$

The mean of the coefficients for all examples are computed and sorted.

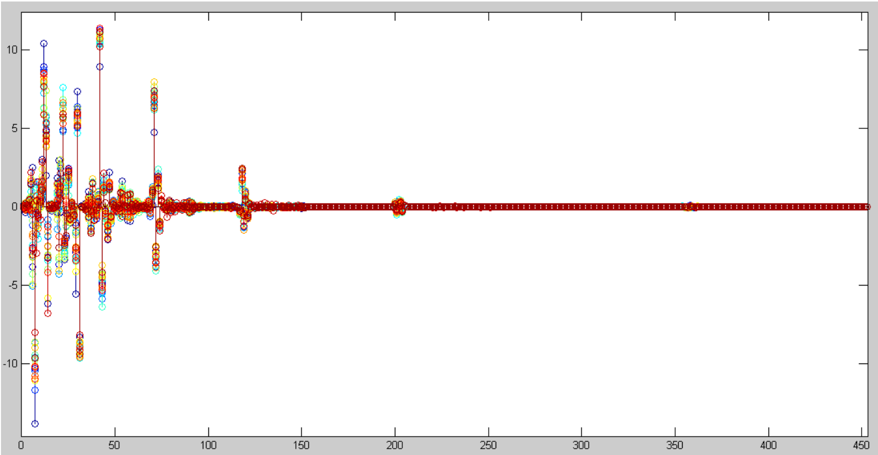


Figure 12: Wavelets coefficients

To represent the signal, the coefficients with most power are computed for each signal. This process is quite long for the training signals but the signal is correctly transformed.

3.2.3 PCA

PCA (Principal Component Analysis) is a method that consists in extracting a base to transform the signal in a new space in which the main energy of the signal is concentrated in the first cooefficients.

The decomposition of the signal $\tilde{s} = \begin{pmatrix} s_1 & s_2 & s_3 \ldots & s_N \end{pmatrix}$ was realized in three steps.

Step 1: Find the autocovariance S and derive the eigenvalues/eigenvectors

$S = \tilde{s}^T * \tilde{s}$

With $\tilde{s} = \frac{1}{\sqrt{N-1}} * (s - \mu)$ µ is the mean of the value of the time points over all the samples used.

The advantage of this matrix is that it is symmetrical so diagonalizable and its eigen vectors are orthogonal $e^T e = Id.$ with $e = \begin{pmatrix} e_1 & e_2 & e_3 & \ldots & e_N \end{pmatrix}$ the eigen vectors.

We can write :

$$S = eDe^T \tag{5}$$

with D the diagonal matrix.

Step 2: Find the components in the new space

To find the eigen vectors and eigen values we remind that $S * e_i = \lambda_i * e_i$

From 5, it can be deduced :

$$\tilde{s}^T * \tilde{s} = eDe^T \Longrightarrow \tilde{s}^T * (\tilde{s}e) = eD = \begin{pmatrix} \lambda_1 e_1 & \lambda_2 e_2 & \ldots & \lambda_N e_M \end{pmatrix} \tag{6}$$

Then, we define c the coefficients in the new space. Indeed, we have :

$$c = \tilde{s}e \; and \; \tilde{s} = ce^T \tag{7}$$

Step 3: Keep the principal components
From 6 it can be deduced :

$$\begin{matrix} s_1 \\ s_2 \\ \vdots \\ s_N \end{matrix} * \begin{pmatrix} c_1 & c_2 & \dots & c_N \end{pmatrix} = \begin{pmatrix} \lambda_1 e_1 & \lambda_2 e_2 & \dots & \lambda_N e_N \end{pmatrix} \tag{8}$$

if we sort the $\lambda_i$we obtain that the last columns are about 0 when lambda is very low. They will be neglected to keep the most important coefficients.

Therefore, the number of eigen values significant M determine the number of coefficients taken into account.

$$\begin{pmatrix} s_1 & s_2 & s_3 & \dots & s_N \end{pmatrix} * \begin{matrix} e_{11} & e_{21} & \dots & e_{M1} \\ e_{12} & e_{22} & \dots & e_{M2} \\ \vdots & \vdots & \ddots & \vdots \\ e_{1N} & e_{2N} & \dots & e_{MN} \end{matrix} = c_1 \quad c_2 \quad c_3 \quad \dots \quad c_M \tag{9}$$

In this case, there are various types of heart beats. Therefore, we use a sample of heartbeats to get the base of eigen vectors. We get the PCA coefficients of the input heartbeats displayed in the figure 13. This figure emphasizes that the power of the signal is concentrated in the first coefficients. After 50 coefficients, the power can be neglected.

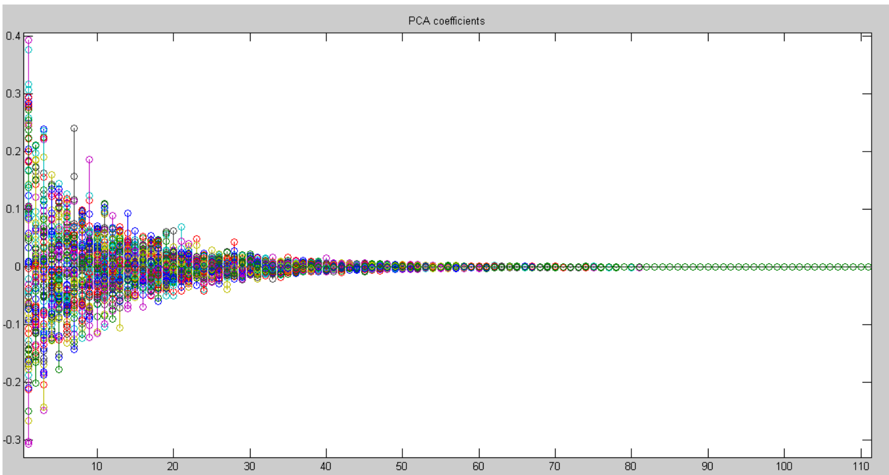
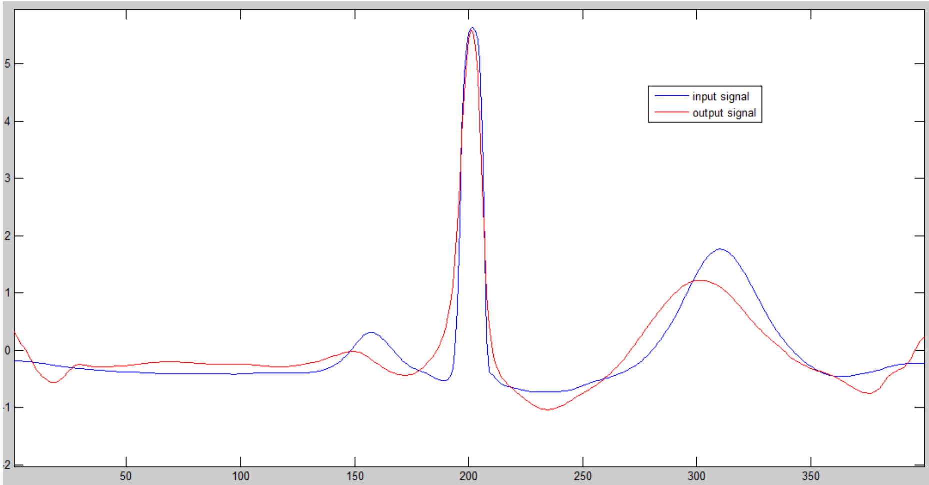
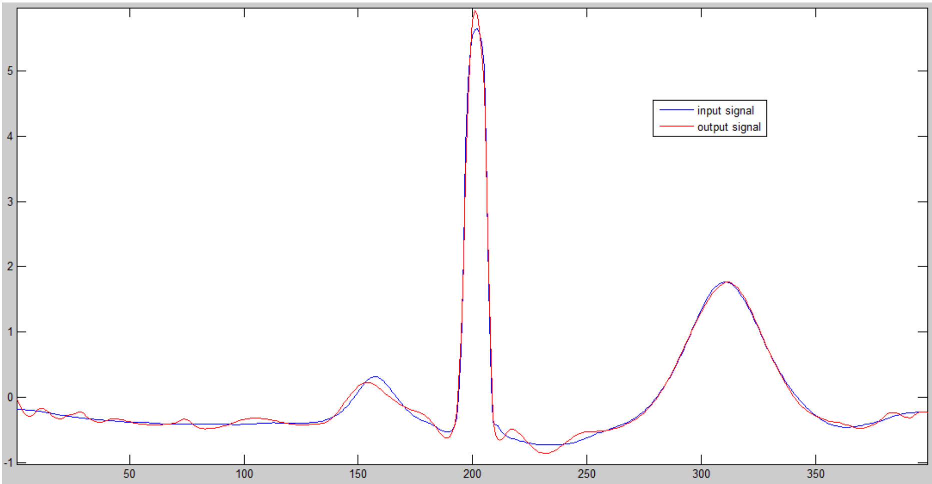


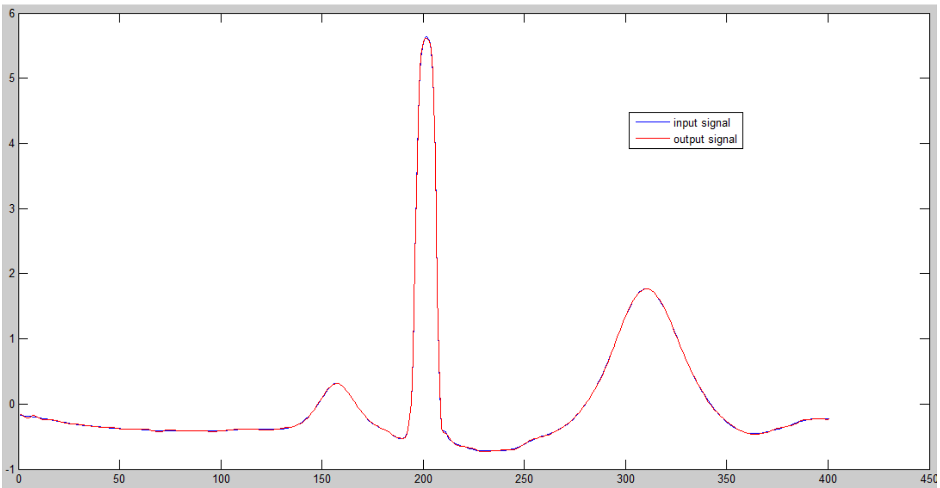
Figure 13: PCA coefficients of 100 beats



(a) Reconstruction with 5 PCA coefficients



(b) Reconstruction with 25 PCA coefficients



(c) Reconstruction with 50 PCA coefficients

Figure 14: PCA reconstruction

In the figure 14, the effect of the coefficients number on the signal representation is emphasized. With only 25 coefficients, we can get the interesting features of the heartbeats. With 5 PCA coefficients, the error in this case is 3.5%, with 25 the error is 0.5% and with 50 coefficients it reaches 0.001%. This method seems to be more efficient than the other to represent the signals. Indeed, the main features are focused in less than 30 coefficients, it is independent from the sampling frequency and it is very quick when classifying. The results in part 5.1 confirms the efficiency of this method.

3.2.4 Other features.

Other features were extracted. First, the dynamic features were used. The R peak are higlighted by most of the annotations, thus, the time intervals were easy to extract from the signals. Three values were used: the interval between the current R peak time and the anterior, the interval between the current R peak time and the following one and then, the actual heart frequency on 16 heartbeats.

To be the closest possible to the doctors analysis, the patient's folder information was included to the input neurons. The header file given with the data base contains basic information. The age and the sexe of the patient were extracted. The idea is to use more pieces of information when creating the specific database of the enterprise for the *Cardian* equipment. The use of these features is evaluated in the section 5.1.

3.3 Neural Networks

3.3.1 Feedforward

It is a basic neural network which is constituted of various layers of computing blocks linked by weights as represented in figure 15.

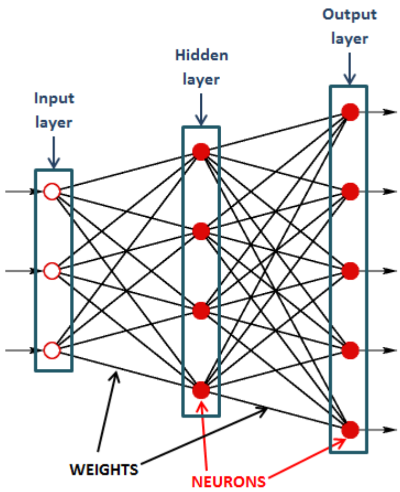


Figure 15: Scheme of a neural network*

**Image taken from the site : http://www.fon.hum.uva.nl*

The process consists in entering the net with the signal features and to get the classification of the signal at the output. In order to do so, classified examples are entered to the net to train it. The weights are varied to adapt the output to the real class expected. The difference measure between the real and the expected outputs can be chosen. In this project, the mean square error (MSE) was used. The main goal is to minimize the result of the sum of MSE for all training examples to get an adapted network. This measure becomes a cost function to minimize called E. Its expression is given in the formula 10. $i = [0, ..., p]$ is the ouput index. $n = [0, ..., N]$ is the training example index.

$$E = \sum_{n=0}^{N} \sum_{i=0}^{p} (c_i - o_i^n)^2 \tag{10}$$

The computing blocks are neurons. They are entities composed of two blocks as it can be seen on the figure 16. o is the ouput of a neuron and the input of the neurons in the following layer. L refers to the L$^{\text{th}}$ layer. Therefore, in the equations above, $J_L$is the number of neurons in the layer L. j is the reference of the neuron in the studied layer. i is the reference of the neuron in the previous layer.

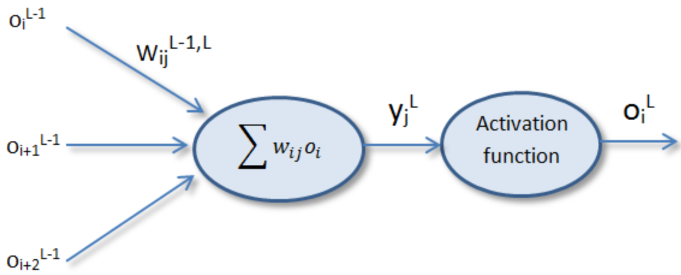


Figure 16: Scheme of a neuron

The first block is commonly a sum of the inputs multiplicated by the weights. The second block is an activation function which can vary. This structure permits to realize a no lineal classification of the inputs. Usely, we can find the sigmoïd, the tanh function or more rarely the radial basis function as in[6]. In this work, the sigmoïd function was used. Its expression is given in the equation 11.

$$f(y) = \frac{1}{1 + exp(-y)} \tag{11}$$

Various algorithms were develop to lower the cost function. They are all based in backpropagation. Backpropagation is an algorithm that consists in computing the derivatives of the cost function with respect to the weights and to deduce a variation to apply to the weights in order to lower the cost function. All the derivatives of the cost function with respect to the weights of the same layer depends on the derivatives of the following layer. The last layer derivatives depends directly on the error between the obtained output and the expected one. In this case, the derivative of the last layer is the following:

$$\frac{dE}{dw_{last-layer}} = (c_i - o_i^n) * o_i * (1 - o_i) \tag{12}$$

For the other layers, the derivative is decomposed in two terms.

$$\frac{dE}{dw_{ij}^{L-1,L}} = \frac{dE}{dy_j^L} * \frac{dy_j^L}{dw_{ij}^{L-1,L}} \tag{13}$$

The first term can be computed with the formulas 14,15,16 and 17.

$$\frac{dE}{dy_j^L} = \frac{dE}{do_i^L} * \frac{do_j^L}{dy_j^L} = \sum_{j=0}^{J_{L+1}-1} \left(\frac{dE}{dy_j^{L+1}} * \frac{dy_j^{L+1}}{do_i^L}\right) * \frac{do_i^L}{dy_j^L} \tag{14}$$

$$\frac{dy_j^{L+1}}{do_i^L} = w_{ij}^{L,L+1} \tag{15}$$

$$\frac{dy_j^L}{dw_{ij}^{L-1,L}} = o_i^{L-1} \tag{16}$$

$$\frac{do_i^L}{dy_j^L} = o_i^L * (1 - o_i^L) \tag{17}$$

The result is expressed in the formula 18.

$$\frac{dE}{dw_{ij}^{L-1,L}} = \sum_{j=0}^{J_{L+1}-1} (\frac{dE}{dy_j^{L+1}} * w_{ij}^{L,L+1}) * o_i^{L-1} * o_i^L * (1 - o_i^L) \tag{18}$$

Therefore, this expression highlights that the output error contained in $\frac{dE}{dy_j^L}$ is propagated backward in the network in order to compute all derivatives.

This algorithm is the base of the neural networks and of this project.

To use these derivatives, various optimization algorithms exist as those listed in [5]. After, having studied the results of these algorithms, it was chosen to use the Gradient descent with momentum rule and the Levenberg Marquardt algorithms. These algorithms will not be described mathematically in this report. The gradient descent depends on two parameters ε and α. The best results were found experimentally for $\varepsilon = 0.005$ and $\alpha = 0.5$. The Levenberg Marquardt algorithm depends on two parameters ν and l. The best results were found experimentally for $\upsilon = 15$ and $l = 100$. The first algorithm is slower than the other. Moreover, the parameters ν and l have less influence on the progress of the algorithm than ε and α. However, the first one is simpler and it has a better memory handling. Indeed, the weights are actualized for each training example whereas in the Levenberg Marquardt algorithm, the weights are actualized after the entry of all training examples.

With this type of neural networks, two structures can be implemented. The first one is a single network with as many outputs as classes. The second one is a structure of as many networks as classes that have one single output. The first one is faster and has a better memory handling. However, each class must present more or less the same training examples number. If not, the calibration of the network turns to be false, the weights are more adapted to one type than another. The second one permit to change this parameter. For each class, the number of training examples is changed depending on how many are available for each class.

3.3.2 Autoassociative

This network developped in[8] uses a similar structure as the previous one and the algorithms used in both of them were the same. The difference lies in the goal of the two networks. The autoassociative networks output isn't the classification but the input itself. One network is trained by class. It is trained exclusively with the heart beats positive for the class. To classify the heart beats, the output is compared to the input and the difference is studied. If the difference is greater than a threshold, the beat is considered negative to the class. In order to do so, the structure of the network is fixed. Five layers were used with repectively, 20, 20, 20 and 30 neurons. It is built as a mix of linear neurons (without activation function) and non linear neurons. However, this process is very heavy and not convenient for the equipment used in this project. Indeed, it aims at managing 7200000 weights.

3.3.3 SOM unsupervised network

This network implemented in [12] has a very different structure. The aim is to map the heartbeat on a 2D map. Depending on where the heartbeat is mapped, the beat is classified as a class of arrhythmias or another one. Therefore, in this configuration, the weights link the input to the map. This structure is shown in the figure 17. The result of the training gives a sorting of the neurons to clarify the limits between classes and to gather the most similar heartbeats. It is a method based on correlation between beats types.

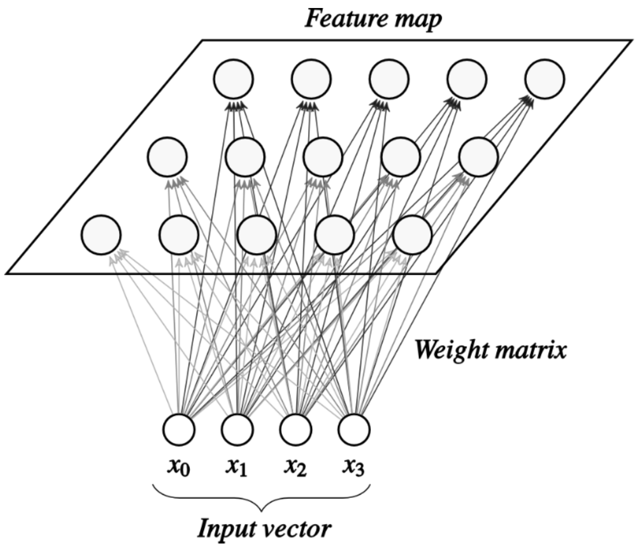


Figure 17: SOM neural networks*

**Image taken from the site : http://gorayni.blogspot.com.ar*

The training algorithm used works as followed :

Step 1: Find the neuron on the map whose the weights are the most similar to the input.

This neuron is called the Best Matching Unit (BMU).

Step 2: Actualize the neighourhood of the neuron selected.

The weights are actualized in the area of the BMU (neuron selected). The most common way is to use the mexican hat function which is represented figure 18.

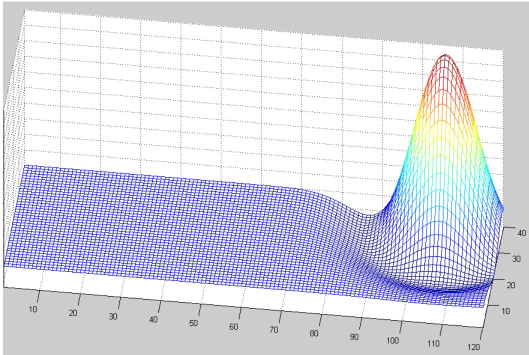


Figure 18: Mexican hat function applied to one neuron

The formula used in this case is expressed in 19.

$$W(d_i) = W(d_i) + \varepsilon * hat(d_i) * (input - W(d_i)) \tag{19}$$

with d_i the distance of the neuron i from the BMU.

Step 3: Determine the spaces of the classes.

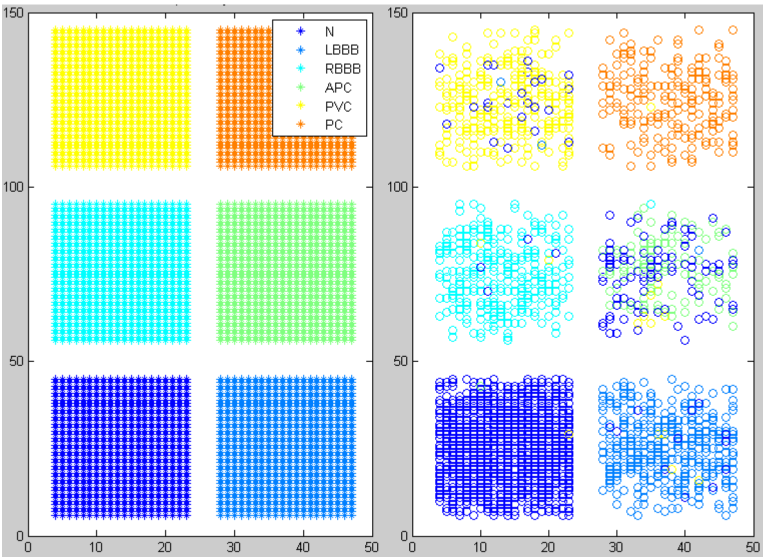
After training the map, a reconstruction is done using the 2D Gaussian function with determinated peak values. This process is discribed figure 20 for the class RBBBB.

First, a simple correlation was done. Indeed, the weights were all preinitialized an no training was done. After, the beats were located on the map and the class was deduced from the location. The results are presented figure 19. In the subfigure 19bOn the right, the initialization is represented. On the left, the result of the evaluation beats mapped on the lattice is represented. It emphasizes the complexity of the classification depending on the class. The Premature Atrial Beats are particularly bad classified. In the other side, the paced beat are very good classified without any training.

	Normal	LBBB	RBBB	PSVC	PVC	PC	unclassified	canal disagree
Normal	3317	28	5	134	32	0	0	250
LBBB	0	462	0	1	5	0	0	6
RBBB	0	0	358	1	0	0	0	2
PSVC	6	1	0	108	0	0	0	6
PVC	3	6	2	7	295	1	0	28
PC	0	0	0	0	0	228	0	0

Accuracy : 98.428% **Specificity :** 99.0998% **Sensibility :** 96.7736%

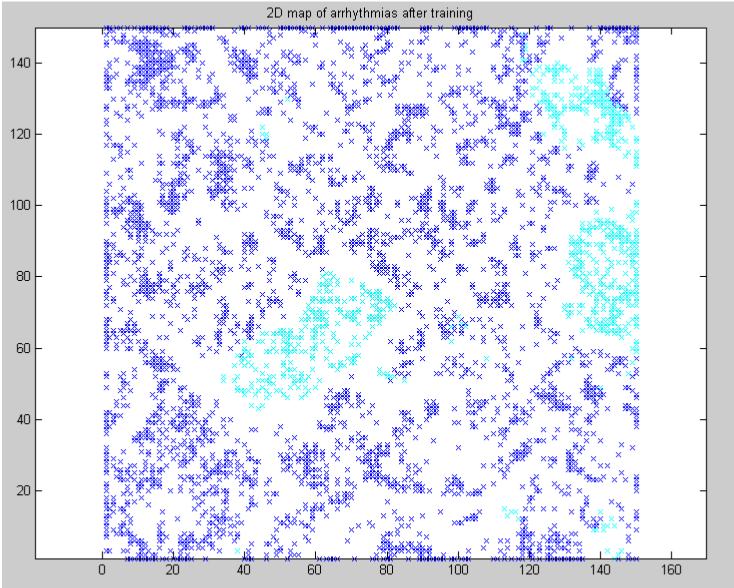
(a) Confusion matrix, accuracy, specificity and sensibility of the method



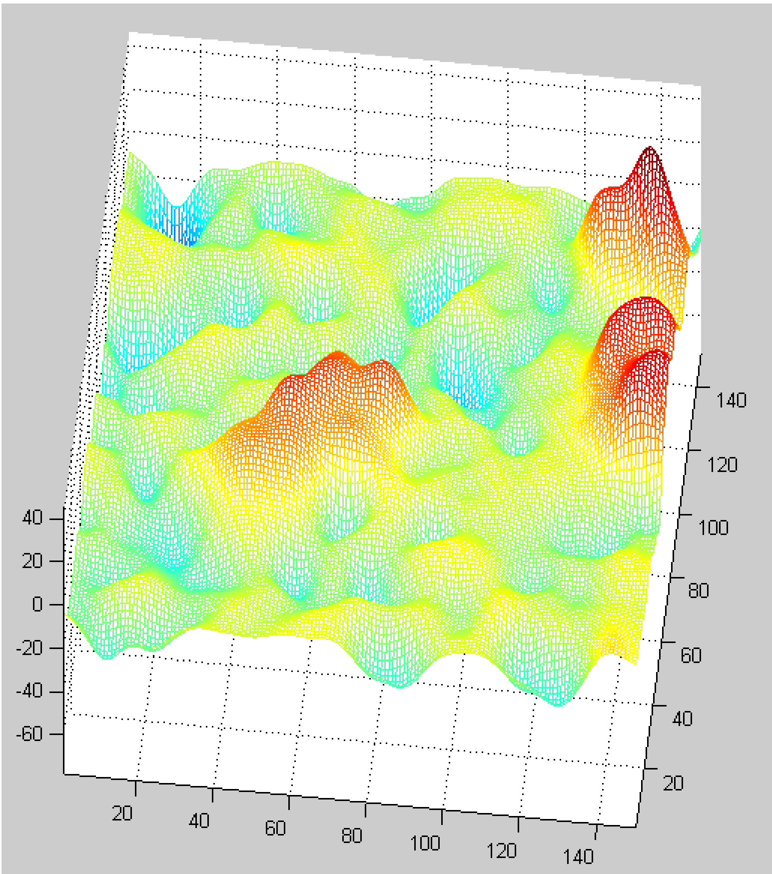
(b) Result of the mapping

Figure 19: Result of the classification by correlation

Secondly, a map was trained without initialization. Due to the difficulty to classify the arrhythmia APC and PVC, it was decided to use maps to classify in two classes for each arrhythmia: positive or negative. After training, the neurons gathered in a logical manner depending on their likelyness. This permits to get a function to classify the hearbeats without giving any information about the classes. In the figure 20a, it can be noticed that the class is well highlighted and a space appears between the RBBB neurons and the others beats.



(a) Map trained



(b) Map reconstruction

Figure 20: SOM training

The results are displayed in section 5.1. It didn't outperform the previous results with a simple correlation.

3.4 Multilead fusion

The output of the network gives a probability of belonging to each class. To evaluate the methods, the class with most probability was selected.

Two leads are available in the MIT BIH data base. In most of the cases, the first lead contains good representations of the normal beats and the second one contains good signals of the arrhythmias classes. In order to use the information of the two leads, different method of fusion were implemented.

a) Better wins

The concept is to keep the result of the derivation which presents the greatest probability.

b) Sum

Here, the sum of the results is computed. The class which presents the greatest sum of probability is selected.

c) Rejection

This method consists in discarding all the beats which lead to a disagreement between channels.

4 Graphic interface

The graphic interface is constituted of three classes included one in the other. This permits to have a progressing analysis of the methods. All the classes are preloaded with initial values to give the user an overview of the range of each parameter and to have datas to test automatically. Advice messages are displayed to make the use easier. It warns the user if the value can't be used or might not be contain in a convenient interval.

4.1 Datas required

Three folders are necessary:
-Functions: it contains three folders, data process, nets and evaluation that correspond to the process of the whole training.
-GUI : it contains the frame classes and the figures of the interface.
-Datos : it contains the initial datas for the frames. Each frame can be run independently.

4.2 Data process

First of all, a window permits to choose the input neurons to enter to a network. The parameters annotated on the figure 21 are the following :

- **1.** The data base to load. This data base must contain in order :
  - Time points.
  - Datas of time intervals between heart beats (commonly the previous one, the following and a mean over some heart beats).
  - Datas of patient information (basicly the sexe and the age).
- **2.** Datas preloaded can be used.
- **3.** The number of samples to treat.
- **4. 5. 6.** The description of the data base (number of classes, number of datas for time intervals and patient information).
- **7. 10.** The kind of transformation to use and the number of coefficients to obtain.
- **8. 9.** The previous feature can lead to the choice of the number of samples to get a transform matrix to get the transform signals (in case of ICA and PCA). The user can choose to enter a rate or a number of samples.
- **11.** The distribution of the set between training and evaluating beats.
- **13. 14.** The use of dynamic features and patient's information.
- **12.** This button permits to process the data base and to pass to the following step.

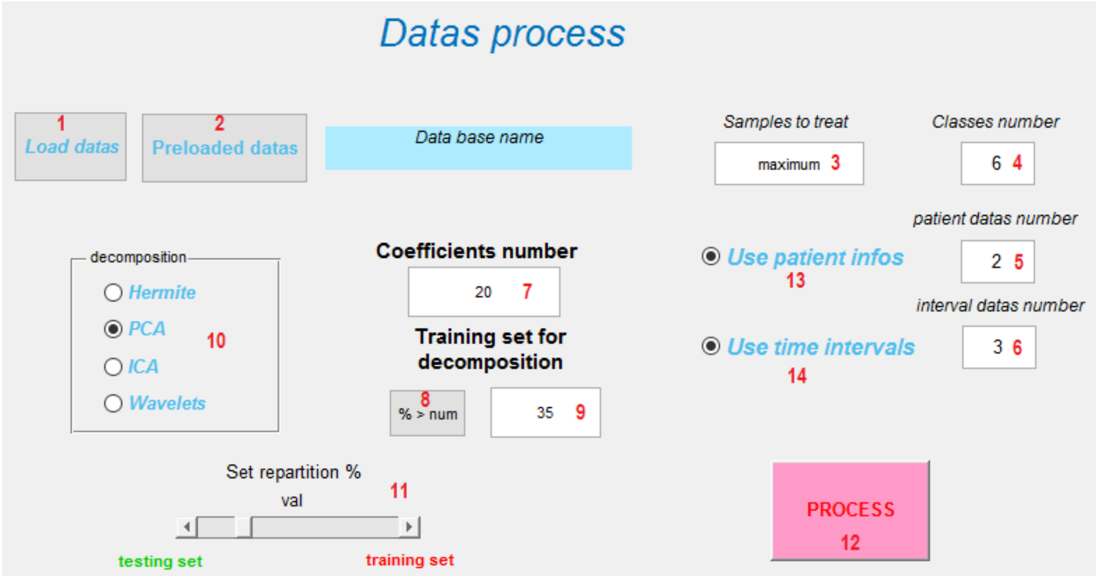


Figure 21: Data process window

This window leads to the following one.

4.3 Network parametrization

This part of the interface permits to initialize the network to use. The parameters annotated on the figure 22 are the following :

- **1.** Learning algorithm to use.
- **2.** Maximum number of iterations.
- **3**. Limit of the cost function to reach.
- **4. 5**. Parameters λ, v for the Levenberg algorithm.
- **6. 7.** Parameters ε, α for the gradient descent. ε is the learning rate, it must be sufficently low to make the approximation be respected and α is the learning momentum that permits to take into account the previous changes in the weights variations.
- **8.** Neurons distance for the SOM algorithm.
- **9.** Use of an autoassociative network.
- **10.** Use or multiply nets (independent on) or multiply outputs (independent off).
- **11.** Structure of the net if a feedforward one is used : a window opens for each layer so the user can enter the number of neurons he wants to put.

- **12.** This button permits to train the net and to pass to the following step.

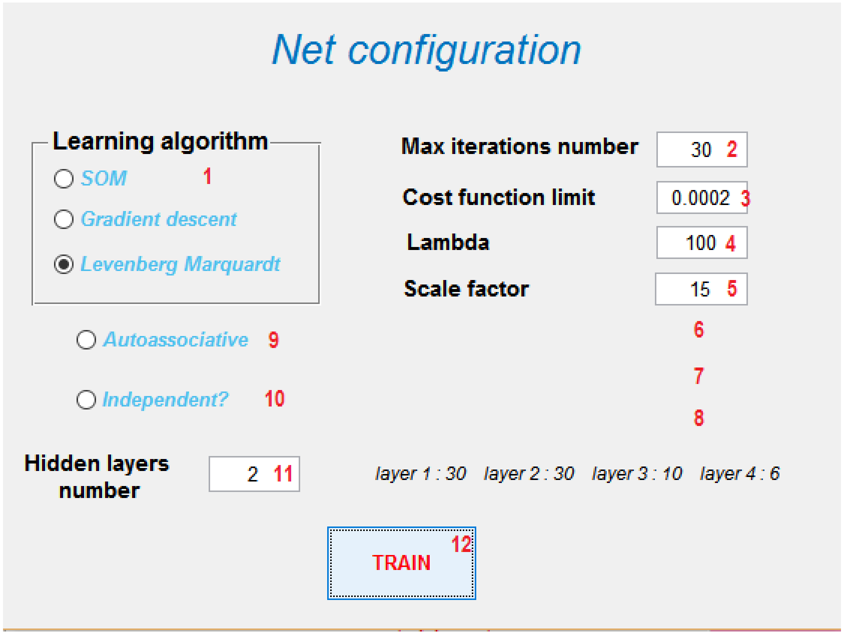


Figure 22: Net parametrization window

4.4 Evaluation type

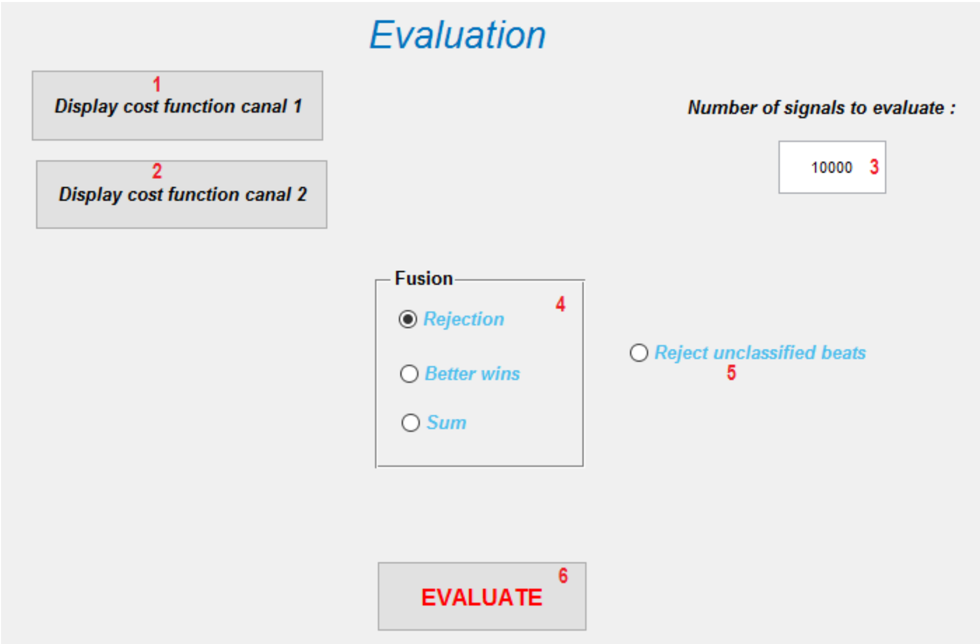
This part of the interface permits to display features of the training and to evaluate the results. The parameters annotated on the figure 23 are the following :

- **3.** Number of samples to evaluate the results.
- **4.** Kind of fusion to use the information of various leads (rejection, better wins or Max(sum) of the two leads).
- **5.** Reject of the beats that are not clearly classified.
- **6.** This button permits to evaluate the net and to display the result.

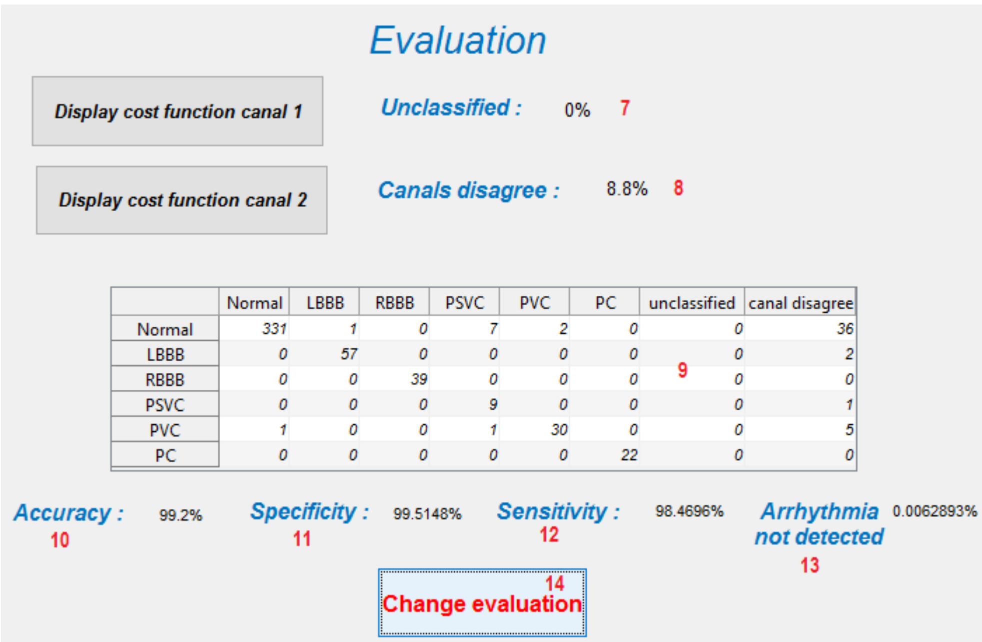
The evaluating tools displayed are the following:

- **1. 2.** Evolution of the cost function during the training to check the good working of the algorithm.
- **7.** Rate of doubtful heart beats.
- **8.** Rate of beats on which the two leads disagree.
- **9.** Confusion matrix. The real nature of the beats are on the left, on rows. The classification is on the right on columns.

- **10. 11. 12.** Accuracy, Specificity and the Sensitivity, they are expressed in the formulas 20, 22 and 21.
- **13.** Number of abnormal beats that are classified as normal ones (dangerous error).
- **14.** This button permits to change the features of the evaluation.



(a) Before evaluation



(b) After evaluation

Figure 23: Evaluation window

5 Results and C implementation

5.1 General results

To compare the results, 30 coefficients for each method were used. Three tools permit to evaluate the methods : the accuracy, the specificity and the sensitivity. They are expressed in the formulas 20, 21, and 22. They are based on the terms True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN) which qualify the classification.

$$accuracy = \frac{TP + TN}{N} \tag{20}$$

$$sensitivity = \frac{TP}{TP + FN} \tag{21}$$

$$specificity = \frac{TN}{FP + TN} \tag{22}$$

All the results presented here were computed evaluating 10000 heart beats.

First, the effect of the input coefficients choice was studied. A single feedforward net was used with a PCA decomposition. The result can be observe in table 2.

	Accuracy (%)	Specificity (%)	Sensitivity (%)
Morphological features	99.0	99.3	96.9
Morphological + dynamic features	99.1	99.5	97.6
Morphological + dynamic features + patient infos	99.2	99.5	97.7

Table 2: Result table

The dynamic features improve significantly the results.

In the following table, the three tool values studied previously are computed for each class and the mean is computed. The evaluation was realized with the maximum of the sum of the two channels as fusion method and all the coefficients available used as input neurons.

		Accuracy (%)	Specificity (%)	Sensitivity (%)	Training time (minutes)
Feedforward single net	Hermite	98.5	99.1	96.9	28
	Wavelets	98.5	99.1	96.4	31
	PCA	99.2	99.5	97.7	11
Independent nets	Hermite	99.5	99.7	97.8	36
	Wavelets	98.2	99.0	94.8	36
	PCA	99.8	99.8	98.4	15
SOM	PCA	98.3	99.0	96.1	23
Correlation	PCA	98.4	99.0	96.8	0.1

Table 3: Result table

These results show that the best method is to use various networks with a PCA decomposition. However, this study will need to be done on the real experimental signals to confirm the results. The training with PCA decomposition is really quicker than with other methods. Moreover, it can be noticed that in all cases the rate of abnormal beats detected as normal ones is negligible (<0.01%).

The type of fusion of datas for the two leads was evaluated too. To do so, we chose to use a single net with PCA decomposition.

	Accuracy (%)	Specificity (%)	Sensitivity (%)
Rejection	99.8	99.9	99.5
Better wins	99.1	99.4	97.3
Max(sum)	99.2	99.5	97.7

Table 4: Result table

This table shows that the most accurate method is to use rejection. However, 8% of the heart beats can't be classified in this case. Therefore, the best method seems to be the third one.

5.2 Best result

The best results were found using the PCA, adding the sexe and the age of the patient and the dynamic feature, a feedforward net with the algorithm Levenberg Marquardt using the following parameters: $v = 15$ and $l = 100$.

The confusion matrix obtained is shown in figure 5.

		Classification					
		N	LBBB	RBBB	APC	PVC	PC
Real classes	N	36886	3	5	140	94	0
	LBBB	0	1148	0	0	2	0
	RBBB	0	0	889	1	1	0
	APC	20	0	0	259	2	0
	PVC	6	0	0	0	801	0
	PC	0	0	0	0	1	608

Table 5: Confusion matrix

The first channel has a rate of good classificaton of 98.85%, the second one of 98.30%. The fusion of the two of them shows a total rate of good classification of 99,32%. This value can reach 99.65% if, for the beats that are not clearly classified (the difference between the probabilities to belong to one class or another is lower than a threshold), the specialist can choose between two classes. The accuracy, specificity and sensibility for each class are detailed in the table 6. It is important to notice that the rate of abnormal beats detected as normal is very low (0.007%).

	N (%)	LBBB (%)	RBBB (%)	APC (%)	PVC (%)	PC (%)
Accuracy	99.35	99.99	99.98	99.60	99.74	99.99
Specificity	99.35	99.83	99.78	92.17	99.26	99.84
Sensibility	99.30	99.99	99.99	99.65	99.75	100

Table 6: Evaluation features

5.3 C implementation

This solution was developped in C in order to make possible the implementation to the equipment of the enterprise. The library ITPP was used and the user manual [2] was studied.

A problem was encountered. Indeed, the Levenberg Marquardt algorithm needs a lot of memory space and multiplication of large matrices. To solve this problem, the library CUDA was used in order to do the process on the GPU. Firstly, a function was developped in order to optimize the process. Secondly, the library CUBLAS was used and gave good time results.

5.3.1 First function

The idea was to do parallel processes to speed up the multiplication. This implies memory transfers to the GPU and loss of time. In this part of the project, a compromise needed to be found between the memory transfers time and the multiplication execution time to make the multiplication faster. The multiplication that spent the most execution time was the following :

$$A^T * A \tag{23}$$

Therefore, the idea was to create a specific function that would just transfer the J matrix from the CPU to the GPU, and realize the multiplication with parallel process. The GPU memory is divided into blocks that contain threads as it is explained in [15]. All the threads are executed at the same time and the threads of a block share memory. The multiplication process is detailed figure 24.

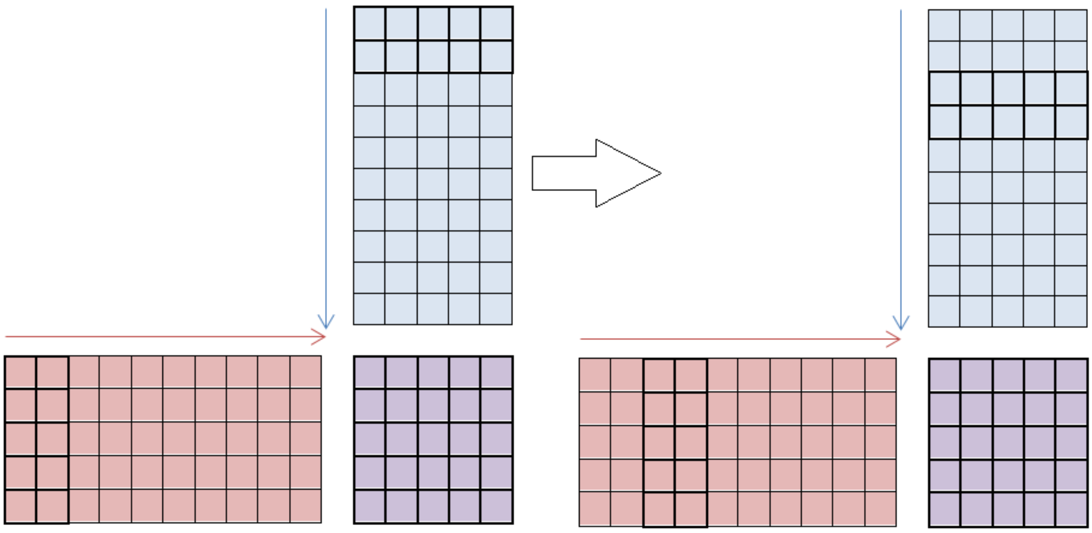


Figure 24: Scheme of the multiplication

All the elements of the result matrix is represented by a thread. The blocks are filled depending on the capacity of the GPU card. Then a loop is run, as shown on the figure 24, to multiply the submatrix that can be contained in the block. In this loop, the result of the multiplication is accumulated in each thread.

This method gives better results than with ITPP. It is efficient to be used for multiplication of matrices from 50x50 to 800x800 values. For larger matrices, it remains quite long as it can be observe on figure7. After studying various libraries available and studying [17], the library CUBLAS was selected to be used in this case.

5.3.2 CUBLAS and results

The CUBLAS library was used and the function sungemm was chosen to realize the multiplication. The times are summarized in the graphic of the figure 7.

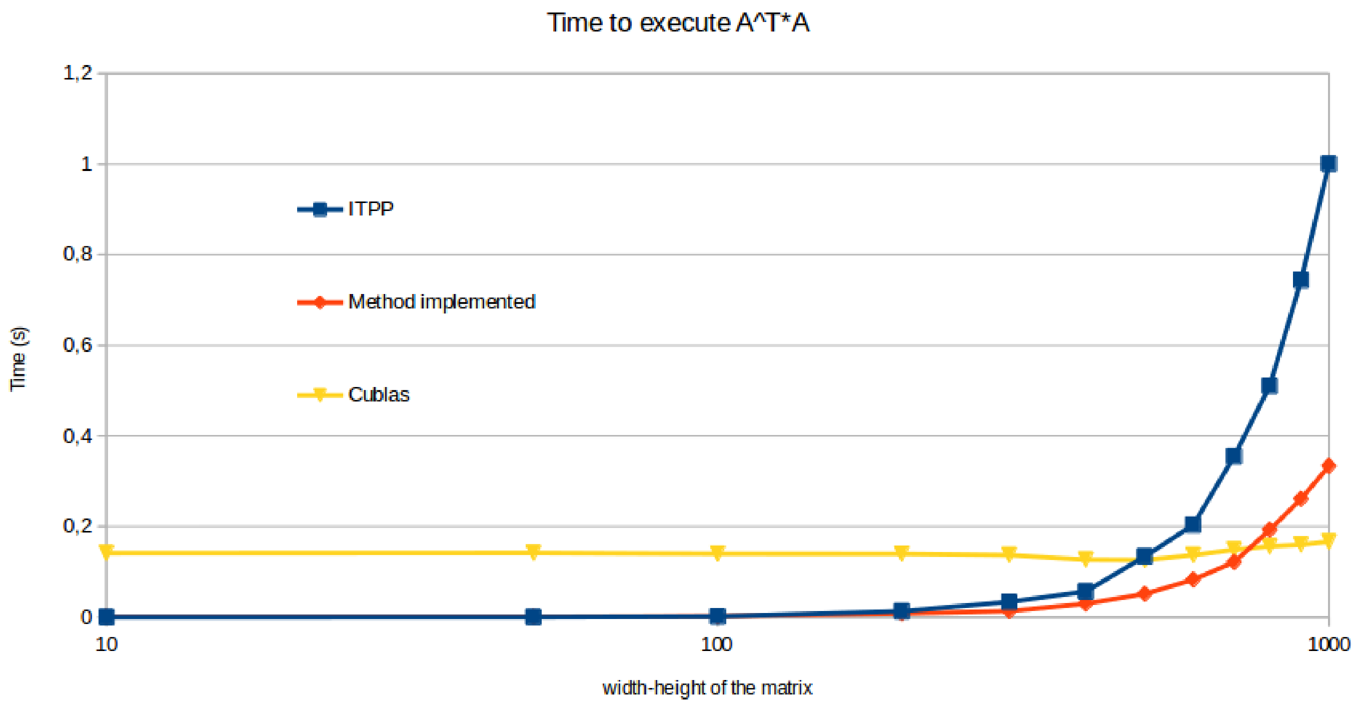
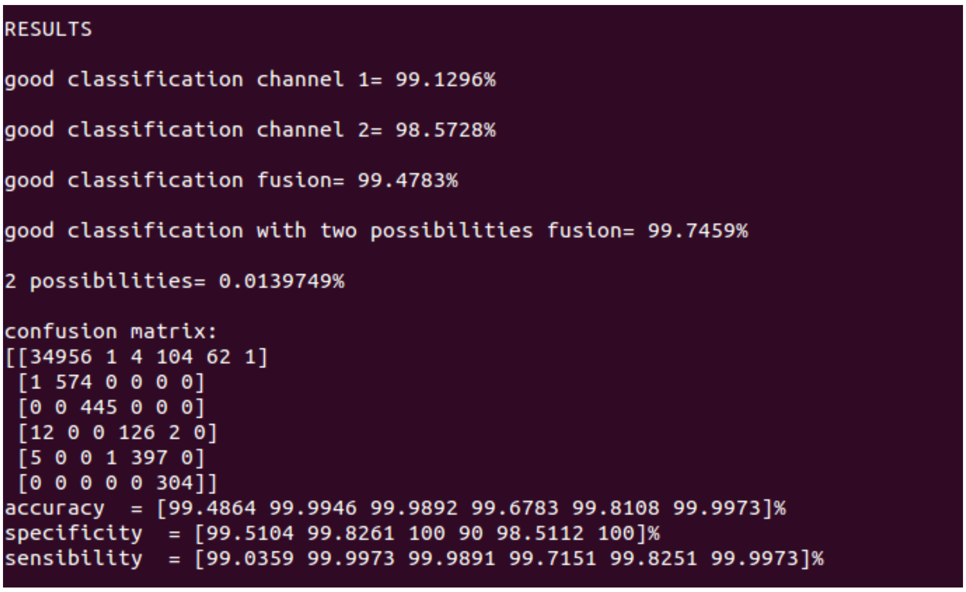


Table 7: Execution time

Therefore, we chose to use the CUBLAS library in the program. The result of the terminal is displayed figure 25.



```
RESULTS
good classification channel 1= 99.1296%
good classification channel 2= 98.5728%
good classification fusion= 99.4783%
good classification with two possibilities fusion= 99.7459%
2 possibilities= 0.0139749%
confusion matrix:
[[34956 1 4 104 62 1]
 [1 574 0 0 0 0]
 [0 0 445 0 0 0]
 [12 0 0 126 2 0]
 [5 0 0 1 397 0]
 [0 0 0 0 0 304]]
accuracy  = [99.4864 99.9946 99.9892 99.6783 99.8108 99.9973]%
specificity  = [99.5104 99.8261 100 90 98.5112 100]%
sensibility  = [99.0359 99.9973 99.9891 99.7151 99.8251 99.9973]%
```

Figure 25: Results displayed on the terminal

The training time is lower than 10 minutes.

6 Conclusion

During this internship, I adquired several skills and general knowledge that will be useful in the future:

• I have now experience in writing Matlab code and Matlab Graphical User Interface.

• I discovered the bases of the working of Linux.

• I also gained experience in reading / writing C++ code, and have a first approach of the ITPP and CUDA libraries.

• I understood the basic principles behind ECG signals adquisition and analysis.

• I improved my habilities at reading papers and finding the information that I need.

I let to the ROEDER enterprise :

• A description of the medical method to establish a diagnosis based on ECG analysis and the main features of the abnormalities encountered.

• A platform with the description of my work.

• A graphical interface to compare classification methods.

• A C program calibrated to classify automatically any beat with any frequency.

• A C program with the training algorithm to include to an equipment in order to pratice the classification with a data base from the Cardian equipment.

I had a great opportunity during this internship to discover an Argentinian enterprise. I was able to see the working of the research and development group and to learn new methods of investigation. I met doctors and participate at meetings to discover new innovating medical products. In the future, I would like to work in the field of image processing and machine learning because of the great variety of areas in which it is implied and because these fields have had a very quick development in the ten last years.

References

[1] D. Davis, "Interpretacion del ECG", *Panamericana*, 1990, 347p.

[2] Pierre Comon, "Independent Component Analysis", *Higher-Order Statistics*, Elsevier, pp.29-38, 1992.

[3] A. Elsayad, "Classification of ECG arrhythmia Using Learning Vector Quantization Neural Networks", *IEEE Computer Engineering & Systems*, 2009.

[4] C. Ye1, M. Coimbra and V. Kumar, "Arrhythmia Detection and Classification using Morphological and Dynamic Features of ECG Signals", *Annual International Conference of the IEEE Engineering in Medicine and Biology*, 2010.

[5] S. Jadhav, S. Nalbalwar and A. Ghatol, "Artificial neural network based cardiac arrhythmia disease diagnosis", *IEEE Process Automation, Control and Computing (PACC),* 2011.

[6] H. Rai, A. Trivedi, S. Shukla, and V. Dubey, "ECG Arrhythmia Classification using Daubechies Wavelet and Radial Basis Function Neural Network", *Nirma University International Conference on Engineering (NUiCONE)*, 2012

[7] V. Nambiar, M. Khalil-Hani and M. N. Marsono, "Evolvable Block-Based Neural Networks for Real-Time Classification of Heart Arrhythmia From ECG Signals", *Biomedical Engineering and Sciences (IECBES)*, 2012

[8] S. Chakroborty, "Accurate Arrhythmia Classification using Auto-Associative Neural Network", *IEEE Engineering in Medicine and Biology Society,* 2013.

[9] A. Ouelli, B. Elhadadi and B. bouikhalene, "Multivariate Autoregressive Modelingfor Cardiac Arrhythmia Classification using Multilayer Perceptron Neural Networks", *IEEE Multimedia Computing and Systems (ICMCS),* 2014.

[10] R. Rodriguez, O. Villegas, V. Cruz Sánchez and J. Bila, A. Mexicano"Arrhythmia disease classification using a higher-order neural unit", *IEEE Conference on Future Generation Communication Technology (FGCT)*, 2015.

[11] S. Kiranyaz, T. Ince, and M. Gabbouj, "Real-Time Patient-Specific ECG Classification by 1-D Convolutional Neural Networks", *IEEE transactions on biomedical engineering*, 2016.

[12] C. Wen, C.H. Huang, M.F. Yeh and K.C. Chang, "Clustering ecg complexes using Selforganizing Cmac", *Biomedical engineeringapplications, Basis & communications*, 2006.

[13] M. Lagerholm and C. Peterson, "Clustering ECG complexes using Hermite functions and Self-Organizing Maps", 2000.

[14] https://www.physionet.org/physiobank/database/html/mitdbdir/mitdbdir.htm

[15] NVIDIA Corporation, https://developer.nvidia.com

[16] ITPP, http://itpp.sourceforge.net

[17] Axel Eirola, "Optimising Matrix Multiplications with CUDA", *Seminar on GPGPU Programming*, 2010.