Optimum Architecture of Neural Networks lane following system

Imen klabi , Afef Benjemmaa , Mohamed Slim Masmoudi , Mohamed Masmoudi
METS Research Group-National Engineers School of Sfax, Tunisia,

ABSTRACT
Recently, neural networks have demonstrated their ability to achieve excellent performance for the control of mobile robots.
In fact, the recourse of this control method by learning has become a necessity because control systems obtain then, proceed by collecting empirical data, storing and removing the knowledge contained in it and using this knowledge to respond to new situations. However, the problem of choosing an optimal number of hidden layers as well as choosing neurons per layer is very critical for these networks. So here we propose to determine the settings for the optimum architecture of neural network.
In the course of our experiments, we have shown that the error of learning as well as the one of the validation provides a satisfactory criterion for the optimization of network architecture.

Keywords
Neural network learning, MLP, MSE, optimum architecture.

1. INTRODUCTION
Neural networks became in few years valuable tools in various fields of industry and services.
Indeed, they possess a remarkable property that is the cause of their practical interest in various areas: they are universal approximators parsimonious [1], and it is in this character that reside the specific nature of neural networks.
In this article, we studied the multi-layer networks that are widely used in the field of robotics [2,3,4].
They may have several hidden layers and it is the way to manage their training which has given renewed interest to their study.
In fact, even if the approximation theorems of the late 80s affirm that, theoretically, a single hidden layer is sufficient to approximate any function sufficiently regular, nothing prevents to implement a back-propagation learning in networks with multiple hidden layers[5].
Besides, some results [5,6] highlight the interest in considering two or more hidden layers to obtain more parsimonious efficient networks and that's by calling several levels of non-linearities.
So the number of hidden layers as well as the number of hidden units in each layer is a very important step to be determined wisely.
In this context lies the purpose of this paper: to determine the optimum architecture of a neural network.

2. DESCRIPTION OF THE LANE FOLLOWING
Our robot has the same configuration as a normal car. Five infrared detectors which are used to find the distance between the vehicle and
its obstacles (see Figure1). Three sensors were placed at the right side of the vehicle facing the parking area and two detectors were installed on the front and rear of the vehicle.

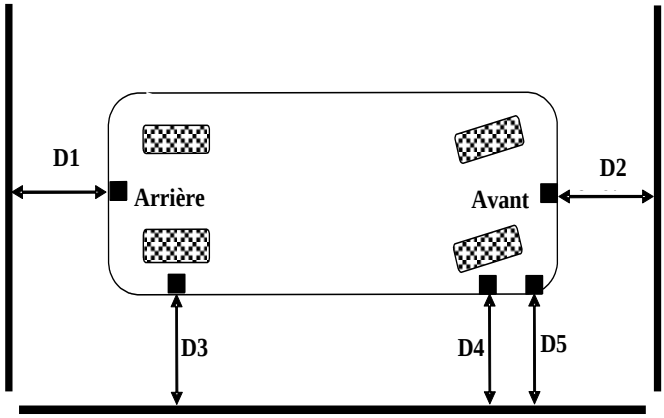


Figure 1: Positioning of the various detectors

3. METHODOLOGY AND EXPERIMENTAL RESULTS
In this paper, a Multi Layer Perceptron (MLP) network with a back propagation algorithm [7] is used.
The input and output of the neuron i, in a MLP mode according to the back propagation algorithm [8] are:

Input: $$x_i = \sum w_{i,j} \times o_j + b_i \quad (1)$$

Output: $$o_i = f(x_i) \quad (2)$$

Where $w_{i,j}$ is the weight of the connection from neurone i to node j, b_i is the numerical value called the biais and f is the activation function.
A back propagation algorithm is designed to reduce error between the actual output and the desired output of the network in the gradient descent manner. The Mean Square Error (MSE) is defined as :

$$MSE = \frac{1}{2}\left(\sum_p \sum_i o_{pi} - t_{pi}\right)^2 \quad (3)$$

Where p indexes the all training patterns and i indexes the output neurons of the network. o_{pi} and t_{pi} denote the actual output and the

desired output of the neuron, respectively, when the input vector, p, is applied to the network.
During training, the weight and biases of the network are iteratively adjusted to minimize the network performance function. Here the performance function for the network is the

mean square error between the network outputs and the target outputs.

The number of hidden units plays a crucial role in controlling the ability of the neural network. In some situations, we may choose the number of neurons in the lowest hidden layer.

At this moment, the network has few parameters and it is likely to have insufficient capacity of learning where it cannot capture all the dependencies that are used to model and predict the values of the observed process.

Contrarily, if we choose a high number of hidden neurons, a problem of over-learning would appear; the network will learn by heart and will give poor results when we will present different data to it.

So we must understand that neural networks are universal approximators, they can model any function if the number of hidden units is sufficient.

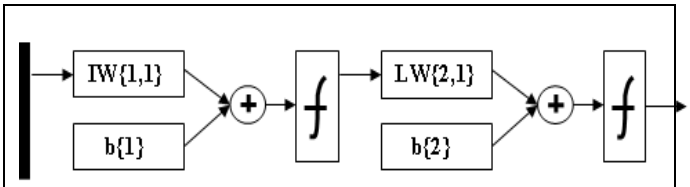
Indeed, to ensure that the neural network sticks to the basic relationships of dependence, we use in addition to all training, a second set called validation set: At the end of each period of training we measure not only the training error but also the validation error, i.e. the total error committed in all the examples of the validation set.

Two types of networks with a single hidden layer and two hidden layers, respectively, are used to determine the optimum architecture.

Having led to a large class of neural networks in matlab, each with a different number of hidden units, we can compare training and validation errors.

3.1 Neural networks with one hidden layer (type I):

Figure 2 presents the first type of network with one hidden layer. Each time we increase the number of hidden neurons in this layer and give the performance progressively.



IW{1,1} is the weight matrix in the hidden layer
b{1} is the biais vector in the hidden layer
LW{2,1} is the weight matrix in the output layer
b{2} is the biais vector in the output layer

Figure 2: The neural networks type I

The number of neurons in this layer varies for NN = (5.9... 25). With NN is the number of neurons in the hidden layer. We got the result shown in table 1.

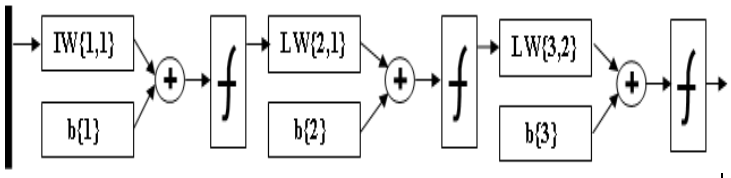
Table 1: Optimal number of neurons in the RN Type I NN = 5, 9, 10,...25

NN	Training performance	Validation performance
5	0.0122	0.2522
9	0.0021	0.2168
10	0.0019	0.2141
11	0.0017	0.2121
12	0.0015	0.1690
13	7.2388e-004	0.1000
14	8.3577e-004	0.2365
15	8.8055e-004	0.1848
16	7.3952e-004	0.2139
17	7.7529e-004	0.2032
20	5.4872e-004	0.2173
25	6.7041e-005	0.2455

Table 1 shows that the performance (MSEt = 7.2388e-004 and MSEv = 0.1000) of the network 5-13-1 are the best among the other networks type I.

3.2 Neural networks with two hidden layer (type II):

Figure 3 shows the second type of network with two hidden layers.



IW{1,1} is the weight matrix in the first hidden layer
b{1} is the biais vector in the first hidden layer
LW{2,1} is the weight matrix in the second hidden layer
b{2} is the biais vector in the second hidden layer
LW{3,2} is the weight matrix in the output layer
b{3} is the biais vector in the output layer

Figure 3: The neural networks type II

The number of neurons varies for:

- 5-NN1-NN2-1 : NN1=3 et NN2 = (3,5,…25),
- 5-NN1-NN2-1 :NN1=5 et NN2 = (3,5,…25),
- 5-NN1-NN2-1 :NN1 = 7 et NN2 = (3,5,…25),
- 5-NN1-NN2-1 :NN1 = 10 et NN2 = (3,5,…20),
- 5-NN1-NN2-1 :NN1 = 12 et NN2 = (3,5,…15),
- 5-NN1-NN2-1 :NN1 = 15et NN2 = (3,5,…10)

NN1 represents the number of neurons in the first hidden layer, NN2 is the number of neurons in the second hidden layer, 5 represents the number of neurons in the input layer and 1 represents the number of neurons in the output layer.
We got the results shown in tables 2-7.

Table 2. Optimal number of neurons in the RN Type II NN1=3 and NN2 = 3, 5, 7,…25

NN1-NN2	Training performance	Validation performance
3-3	0.0171	0.2413
3-5	0.0047	0.1740
3-7	0.0045	0.0590
3-10	0.0028	0.0464
3-15	0.0013	0.2391
3-20	8.3059e-004	0.2726
3-25	5.0371e-004	0.2080

Table 2 shows that the performance (MSEt = 0.0028 and MSEv = 0.0464) of the network 5-3-10-1 are the best.

Table 3. Optimal number of neurons in the RN Type II NN1=5 and NN2 = 3, 5, 7,…25

NN1-NN2	Training performance	Validation performance
5-3	0.0038	0.1879
5-5	0.0022	0.2345
5-7	9.1175e-004	0.1731
5-10	8.8884e-004	0.0668
5-15	2.5297e-004	0.1874
5-20	9.1206e-006	0.2014
5-25	1.2411e-006	0.2097

Table 3 shows that the performance (MSEt = 8.8884e-004 and MSEv = 0.0668) of the network 5-5-10-1 are the best.

Table 4. Optimal number of neurons in the RN Type II NN1=7 and NN2 = 3, 5, 7,…25

NN1-NN2	Training performance	Validation performance
7-3	0.0026	0.2293
7-5	0.0028	0.2262
7-7	4.6031e-004	0.2218
7-10	9.4999e-005	0.2095
7-15	1.9471e-005	0.2064
7-20	6.3533e-005	0.2053
7-23	1.4151e-005	0.1067
7-25	6.4267e-009	0.2002

Table 4 shows that the performance (MSEt = 1.4151e-005 and MSEv = 0.1067) of the network 5-7-23-1 are the best.

Table 5. Optimal number of neurons in the RN Type II NN1=10 and NN2 = 3, 5, 7,…20

NN1-NN2	Training performance	Validation performance
10-3	0.0048	0.1335
10-5	2.0817e-004	0.0762
10-7	1.8126e-004	0.0534
10-8	1.4334e-005	0.1934
10-10	4.5780e-006	0.2271
10-15	3.6188e-009	0.2379
10-20	1.7309e-010	0.2859

Table 5 shows that the performance (MSEt = 1.8126e-004 and MSEv = 0.0534) of the network 5-10-7-1 are the best.

Table 6. Optimal number of neurons in the RN Type II NN1=12 and NN2 = 3, 5, 7,…15

NN1-NN2	Training performance	Validation performance
12-3	3.0091e-004	0.1420
12-5	2.2950e-005	0.1171
12-7	6.3429e-006	0.1888
12-10	4.2428e-006	0.1518
12-15	6.2401e-011	0.1943

Table 6 shows that the performance (MSEt = 4.2428e-006 and MSEv = 0.1518) of the network 5-12-10-1 are the best.

Table 7. Optimal number of neurons in the RN Type II NN1=15 and NN2 = 3, 5…10

NN1-NN2	Training performance	Validation performance
15-3	1.9579e-005	0.2315
15-5	2.0981e-006	0.2017
15-7	5.5098e-007	0.1912
15-10	3.5035e-009	0.2528

Table 7 shows that the performance (MSEt = 5.5098e-007 and MSEv = 0.1912) of the network 5-15-7-1 are the best.

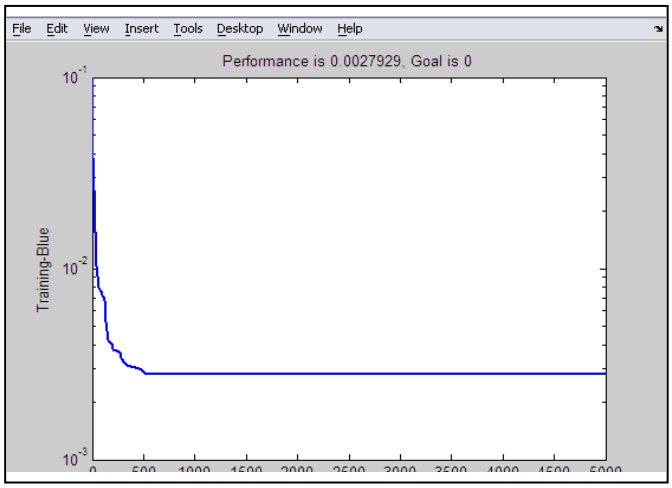
3.3 Interpretation results

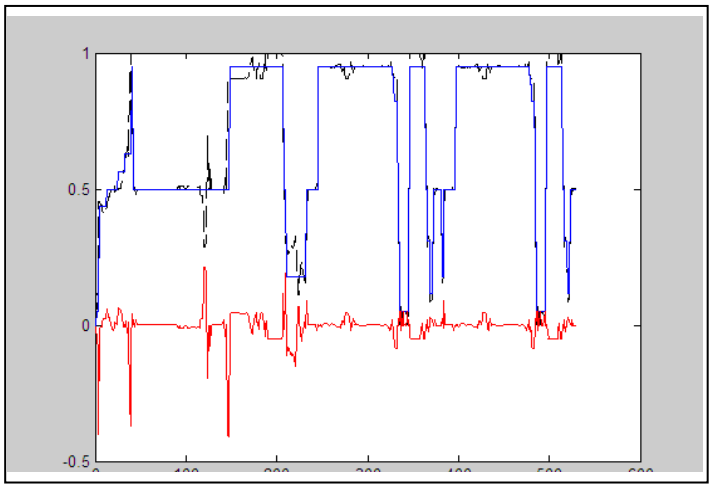
We notice that the training error decreases progressively as the number of hidden units increases.
The validation error, in its turn, is high when the number of hidden units is low, decreases with increasing the number of hidden units, reaches a minimum for an optimal number of hidden units, and increases when the number of units becomes too large.
So the use of a validation set, distinct from the training set, allows us to choose the optimal number of hidden units or neurons. In some situations, we find that the validation error may increase when the error of learning decreases too much. We say then, that the neural network suffers from over-learning.
After these test simulations in Matlab, the optimum architecture of our neural network is chosen according to these two criteria: training performance and validation performance.
Table 8 shows the best results of network architectures trained.

Table 8. Summary table of the best architecture networks

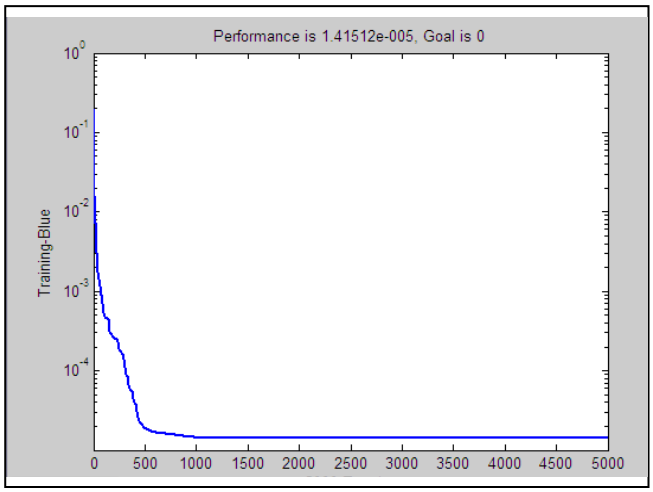
NN	Training performance	Validation performance
13	7.2388^{e}-004	0.1000
3-10	0.0028	0.0464
5-10	8.8884^{e}-004	0.0668
7-23	1.4151^{e}-005	0.1067
10-7	1.8126^{e}-004	0.0534
12-5	2.2950^{e}-005	0.1171
15-7	5.5098^{e}-007	0.1912

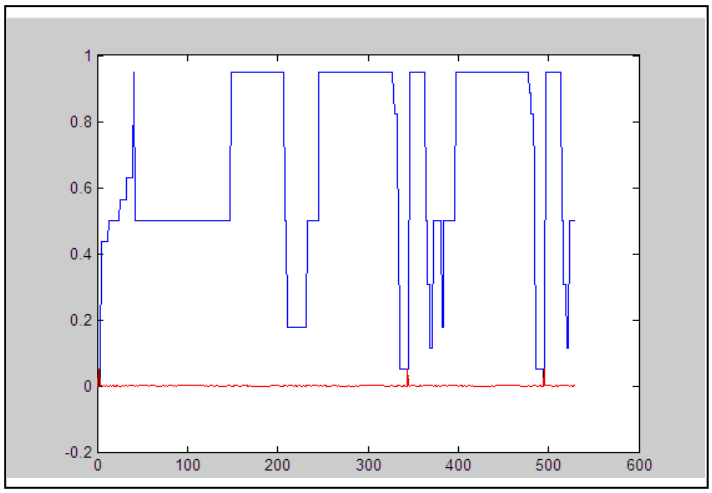
As final result, the optimal architecture of our neural network is constituted of 10 neurons in the first hidden layer and 7 neurons in the second hidden layer.
Figure 4 (see end of the paper) show the graphs of the learning performance (left graph), variation at the desired output and the output of the network and look of the error (right graph), for some networks.



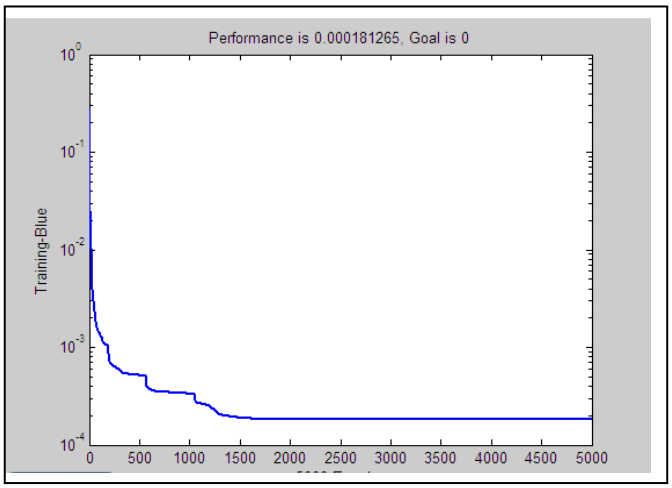


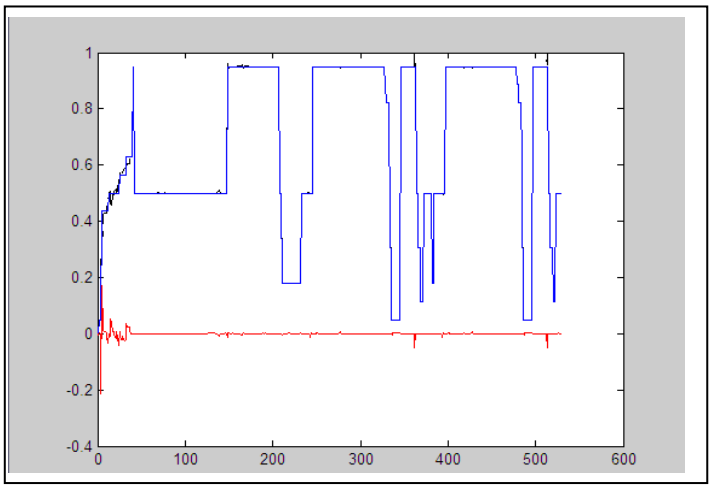
5-3-10-1



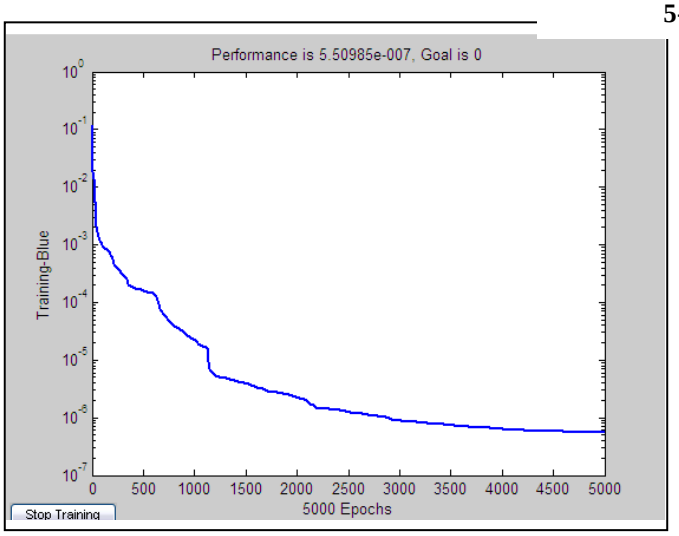


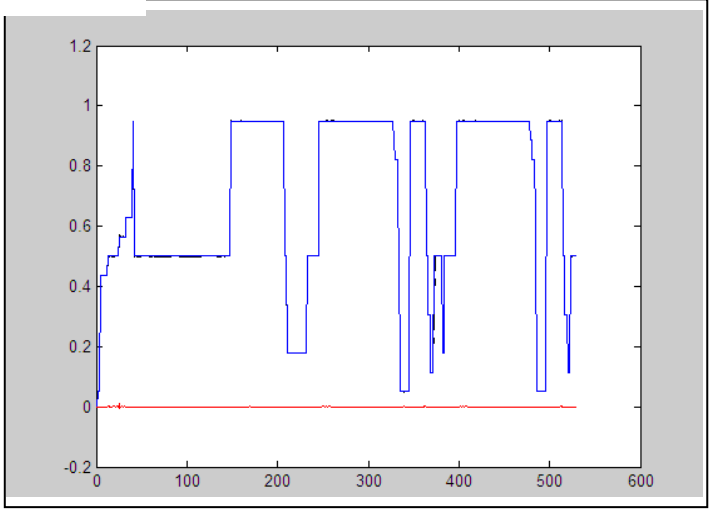
5-7-23-1



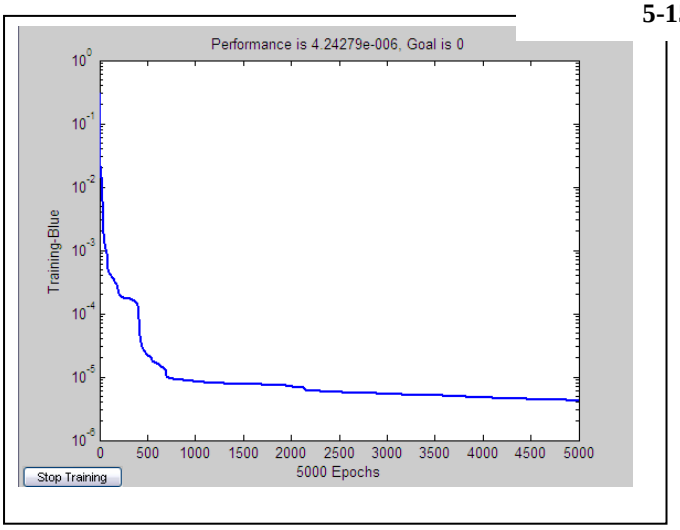


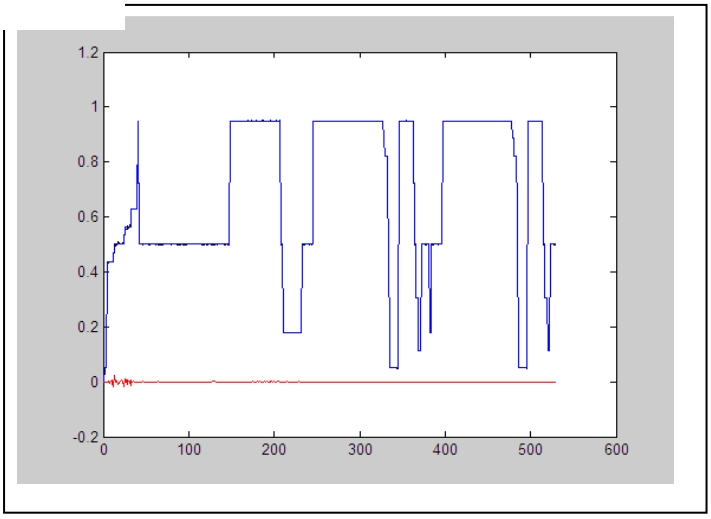
5-10-7-1





5-15-7-1





5-12-10-1

Figure 4: Pace of learning performance (left graph) and variation at the desired output and the output of the network and look of the error (right graph).

4. CONCLUSION

The choice of architecture is an important task for the implementation of neural network, (external data, the number of hidden neurons, and arrangement of these neurons) so that the network is able to replicate what is deterministic in the data.

The number of adjustable weights is one of the fundamentals of a successful application. Indeed, the problem of determining the optimum architecture remained for a long time an open problem.

In this article, we have sets of tests that determine the architecture for a wide class of networks with learning based on back propagation algorithm.
So, it is on the basis of error training and the one of validation that we managed to choose the optimum architecture.

5. REFERENCES

[1] Sana Ben Romdhane, 2010, « Robots d'assistance pour les personnes handicapées : commande linéarisante par retour d'état et neuronale d'un système holonome à mise en position continue» Mastère, INSAT.

[2] Mark Buplawsky et Michal Bialko, May 2008, « Implementation of Parallel Fuzzy Logic Controller in FPGA Circuit for Guiding Electric Wheelchair », Article, Department of Electronics and Computer Science, Koszalin University of Technology, Koszalin, Poland.

[3] Hoang T. Trieu et Hung T. Nguyen, August 2008, « Advanced obstacle avoidance fora laser based wheelchair using optimised bayesian neural networks », international IEEE EMBS conference, British Coulombia, Canada.

[4] Nghia Nguyen et Hung T. Nguyen**,** septembre 2009, « Robuste Multi-variables strategy and its application to a powered wheelchair », international IEEE EMBS conference, Minneapolis, Minnesota, USA.

[5] Ludovic Arnold, Hélène Paugam-Mois Y et Michèle Sebag, 2010, « optimisation de la Topologie pour les Réseaux de Neurones Profonds », Article, Université Paris-Sud et Université de Lyon.

[6] Mohamed Slim Masmoudi, avril 2008, « Conception et implémentation de systèmes intelligents appliqué au stationnement d'un véhicule-robot », thèse de Doctorat, ENIS.

[7] Fei Jiang, Decembre 2009, «Optimisation de la Topologie de Grand s Réseaux de Neurones », these de Doctorat, Université Paris-SUD.

[8] J.Amini, Decembre 2008, «Optimum Learning Rate in Back-propagation Neural Network for classificationof Satellite Images (IRS-1D) », Article, Scientia Iranica, vol.15, NO.6, PP.558-567, Sharif university of Technology.