NEURAL TECHNIQUES FOR ENERGY MANAGEMENT IN BUILDING

F. Garzia, G.M. Veca
Università di Roma "La Sapienza"

ABSTRACT

A versatile system to manage and control the energy in building is presented. It is based on neural technology to adapt its management strategy to the controlled environment. The neural techniques adopted is properly chosen to be implemented in a low computation capabilities device such a common electronic microcontroller.

1. INTRODUCTION

The management and the control of the energy flows inside a building can be made using different system architectures [1-4]. They differ for the features and performances and obviously for the cost necessary to install them [5-7].

Once chosen a particular hardware platform [8] it is necessary to implement a proper software that implements and executes the desired energy management strategy.

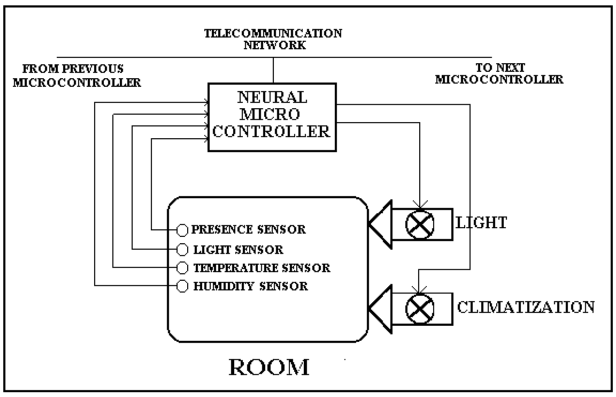
The choice of the energy strategy needs to know 'a priori' the exigencies of the final users together with their energy consumption time table, that is a certain number of data must be collected, usually, for a long time.

This problem can be avoided using adapting strategies such as the one offered by the neural networks [9-11].

Neural networks offer the great advantage of learning the behaviour of the final users, together with a great generalisation capability that make them able to face new situations adopting a mixed behaviour between the learned ones.

The input can be represented, for example, by the presence of people inside a room, the outside temperature, the inside temperature, the time, the date

Fig.1 Example of general scheme of the controlling



architecture

and other data that are useful to characterise the desired application and so on.

The output data are represented by the desired energy management strategies as a function of the input data that act directly on the electrical and the air conditioner installations.

Different neural networks architectures can be used to achieve the desired purpose, each characterised by peculiar features: a neural architecture is more performing with respect to the others as a function of the number of inputs and their relation with output data.

As a general law it is possible to state that the greater is the complexity of the desired energy strategies and the higher is the number of elementary cells (neurones) that compose the neural network: since the net needs an initial training time, that strongly depends on its complexity, the higher is the number of neurones and the longer is the training time.

Different learning algorithms can be used to train the neural network, each characterised by particular features: a basic law that can be stated is that the higher is the precision of the learning algorithms and the slower is the learning velocity, that generally ensures a converge towards the correct desired output.

Further it is possible to use an adapting strategy during the normal working, that is the net is trained to face new situations. This on-progress adapting is recommended only if the computation duties of the controller module are not so heavy with respect to its capabilities, to avoid to slow its work, even if the velocities of variations of the energy phenomena are quite slow.

Since the computation resources of the electronic module that controls the input data and the output devices are unavoidably limited, it is necessary to reduce as more as possible the number of input data and the complexity of the energy management strategy, that is find the most suitable neural networks for this kind of application, which is the purpose of this paper.

2. THE NEURAL NETWORKS

Neural networks find actually a lot of applications in different fields such as:

1) electronics: process control, machine vision, voice synthesis, linear and nonlinear modeling, signal analysis;
2) robotics: trajectory control, vision systems, movement controller;
3) telecommunications: image and data compression, noise reduction,
4) security: face recognition, voice recognition and other biometrics applications, new sensors;
5) defense: weapon steering, signal and image identification, radar and image signal processing, object discrimination and recognition;

and other fields such as aerospace, insurance, banking, manufacturing, automotive, medical, financial, entertainment.
The common element of their field of applications is the need of classifying a given element as belonging to one or more given classes.
One of the main referring model for the reproduction of human intelligence is the so called ‘Connectionism’ that postulates the logic equivalence between any structured knowledge and a proper neural network. The Connectionism allows to develop a new form of artificial intelligence based on a sub-symbolic computation instead of the symbolic computation that represents the typical application field of the classical artificial intelligence. The Connectionism originates from the study of the working mechanisms of the central nervous system of biological organisms.
Human brain is composed by neurons that are cells whose purpose is represented by the information processing. Each neurons is connected with the other by means of a central body called axon and by numerous terminations called dendrites. The connection points between neurons are called synapses that show an excitatory behavior if they allows the electrical pulses to pass or an inhibitory behavior if they stops these pulses.
Each neurons behaves as an adder of the pulses generated by nearby neurons: if the sum overcomes a certain threshold the neuron actives letting the information to proceed along its path.
The connections between neurons can be modified allowing the memory effect to take place.
Artificial neural networks imitate this mechanism generating a knowledge database by means of the modification of the connections of a net that can learn from direct experience modifying its internal state to adapt to the solution of a particular problem.
The modeling of the behavior of neural networks is quite complex and generally uses the approach of the dynamic systems and the related concepts such as cycles, strange attractors and equilibrium points.
Neural networks are particularly useful when the law related to a certain phenomenon is not known in a deterministic way but it is necessary to reproduce it.
Neural networks are very useful when:

1) it is necessary to generalize the knowledge acquired on a restricted base to a wider base;
2) a certain situation changes with time;
3) data are not complete, uncertain or influenced by errors;
4) it is necessary a great tolerance to troubles or misfunctions;
5) it is necessary to find rapidly a heuristic solution to a particular problem;
6) a phenomenon rapidly changes and short adapting times are requested;
7) a high computational parallelism is requested;
8) a proper algorithm is not known;
9) qualitative or incomplete data are present;
10) the problem is data intensive instead of number crunching;
11) it is necessary to produce a knowledge for an expert system.

For all these reasons neural networks represent a useful and flexible tool for a lot of situations.

2.1 NEURAL NETWORKS MODELING

The elementary computation element of this kind of technology is represented by the neuron, that is a cell that receives one or more input values and produces one or more outputs that depend on the input values.

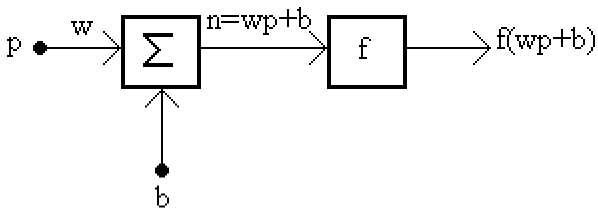


Fig.2 Single input neuron

Considering a single input - single output neuron, if p is the input and the w the weight, the product wp reaches the Σ unit where it is summed to a bias value b, that can be considered as an input of value equal to 1 and whose weight is equal to b. The computed quantity n=wp+b reaches the transfer function f that calculates the output of the neuron a=f(wp+b).
The parameter w,p and b are adjustable and they can be adapted so that the neuron exhibits an interesting or desired behavior.
Therefore an elementary neuronal cell performs simple operations such as addition and multiplication which can be easily executed by low computation capabilities devices. Their strength relies on their organization in massively parallel architectures.
The elementary cells can be connected in different way to form a neural net that can be trained to do a particular job adjusting properly their weights and/or their biases (supervised learning) or letting it to learn by itself (unsupervised learning that is typical of the self organizing nets).
The transfer function of the neuron can have different expressions that are: step (symmetric and asymmetric), linear (with saturation or without saturation), sigmoidal (logarithmic or tangential), triangular, radial and others.

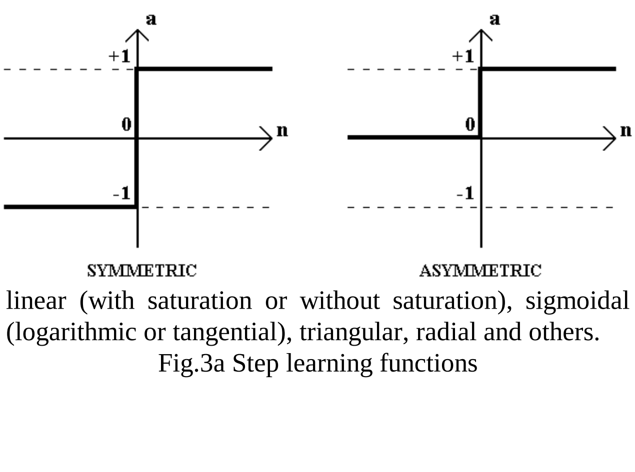


Fig.3a Step learning functions

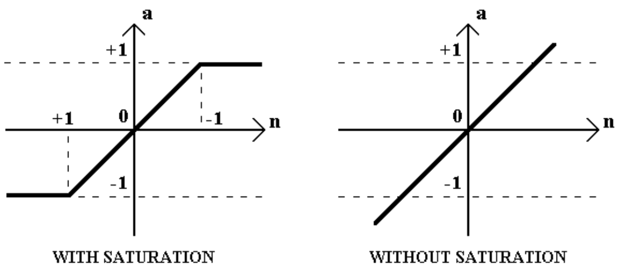


Fig.3b Linear learning functions

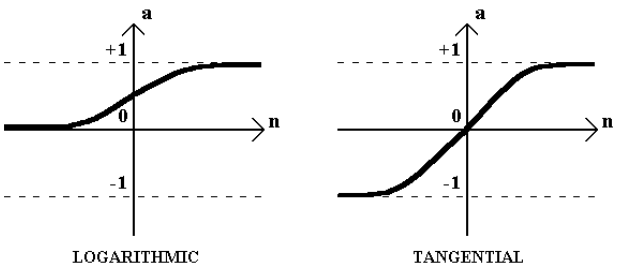


Fig.3c Sigmoidal learning functions

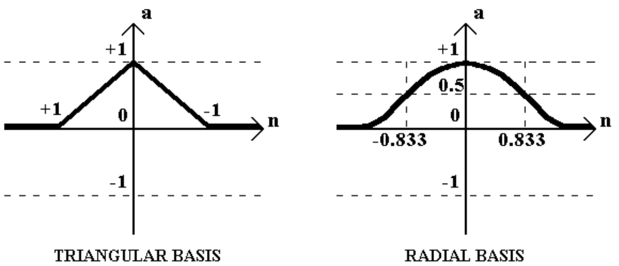


Fig.3d Limited learning functions

If the neuron has more inputs:

$$\mathbf{p} = [p_1, p_2, ..., p_R] \qquad (1)$$

each of them is multiplied by the weights:

$$\mathbf{w} = [w_{1,1}, w_{1,2}, ..., w_{1,R}] \qquad (2)$$

and the sum unit executes the dot product **wp**, adding the bias b to give:

$$w_{1,1}p_1 + w_{1,2}p_2 + ... + w_{1,R}p_R + b \qquad (3)$$

that is the argument of the output transfer function.

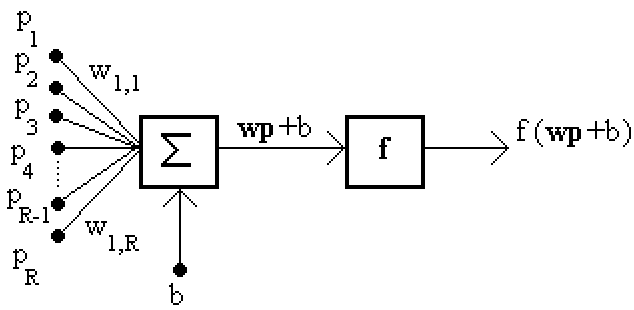


Fig.4 Multiple input neuron

Two or more multiple neurons can be combined to generate a layer of neurons. Considering a layer composed by S neurons, each element of the input vector **p**, composed by R elements, is connected to each neuron input through the weight matrix **w**. The j-th neuron weights properly its inputs, performing a dot product and adding the j-th bias to generate its scalar output n(i). The various values n(i) taken together, form a vector **n** composed by S elements. Each element of the vector **n** represents the input of the transfer function of the relative neuron. At the output a column vector **a** is obtained.

Generally the number of inputs R is different from the number of neurons S.

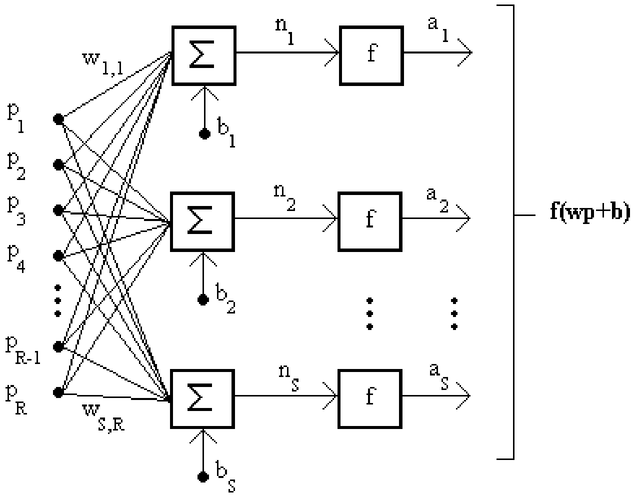


Fig.5 Layer of neurons

In a layer of neurons the weight matrix **w** has the following form:

$$\mathbf{w} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,R} \\ w_{2,1} & w_{2,2} & \cdots & \\ & & & \\ w_{S,1} & & \cdots & w_{S,R} \end{bmatrix} \qquad (4)$$

where the row indices indicate the destination neuron of the weight and the column indices indicate which source is the input for that weight. For example, the weight labeled with (3,2) expresses the strength of the signal from the second input element to the third neuron.

When we deal with a multiple layer net we deal with different weight matrixes **w**, different bias vectors **b**, and different output vectors **a** each of them referring to the relative layer.

In this situation the first layer is called input layer, the network output is called output layer and the intermediate layers are called hidden layers.

Multiple layers nets can perform complex functions. For example a two layers net, where the first layer is sigmoid and the second layer is linear, can be trained to approximate any function with a finite number of discontinuities.

Networks with biases are able to represent relationships between inputs and outputs easier than networks without biases. In fact a neuron without a bias will always have a net input to the transfer function equal to zero when all of its inputs are zero while a neuron with bias can learn to have any net transfer function net input under the same conditions by learning an appropriate value for the bias.

2.2 NEURAL NETWORKS LEARNING

We already said that a neural network can be trained to do a particular job adjusting properly its weights and/or its biases (supervised learning) or letting it to learn by itself (unsupervised learning that is typical of the self organizing nets).
We only consider the supervised learning that is the method used for the kind of net that is necessary for our purpose.
In this case an input is presented to the net and the weights and biases are updated until the net gives as output the desired output or almost an approximation with a controlled error.
The learning algorithm strictly depends on the kind of net: for each net different learning algorithms are generally available and each of them shows a particular features such as velocity, precision, low memory occupation, etc.
Anyway the net can be trained in an incremental mode or in a batch mode. In the incremental mode weights and biases are updated any times an input is presented at the net while in the batch mode they are update only when all the inputs are presented at the net. The difference between the two methods is that in the incremental mode the net tends to forget the behavior it must show for the first inputs it has learned with respect to the last inputs, since the weights and biases are continuously updated to respect the requests of the more recent inputs. In the batch mode, on the contrary, the weights and biases are updated to respect the desired behavior for the different inputs all together. For this reason, when all the inputs and the desired outputs are available at the same time, batch mode has to be preferred with respect to the incremental mode.
In the situation that we are going to consider the inputs are represented by the state of occupation of a particular room by the user that is therefore not available all at once unless a long and unpractical period of observation of the user has been made. Since we consider the net to start to work directly on the field, we want it to learn in real time, that is to learn the user behavior while it works and therefore the inputs are not available at the same time, leading us to train the net in an incremental mode.

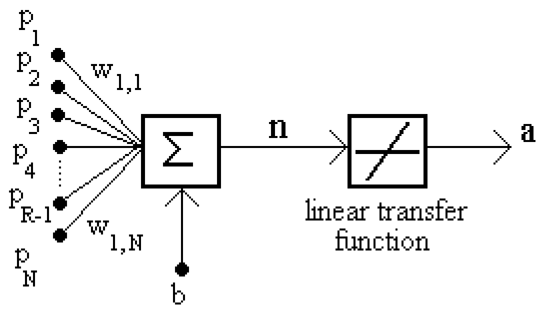
2.3 DEFINITION OF THE PROBLEM

Different parameters such as presence, light intensity, temperature, humidity, etc., can be used as inputs to the neural network to control the energy flows, using nets characterized by a growing complexity. Since we use as hardware a microcontroller with quite limited computation resources, it is necessary to reduce as more as possible the complexity of the net to reduce the computation duty and increase its velocity.
For this reason we decide to use as input only the presence of people inside the room, that is a binary information, and as output the switching on/off of the electrical loads, that is a binary information too.
The neural net has therefore to learn the occupation state of the controlled room, that is to predict when it is possible and necessary to switch on or off the electrical loads as a function of the previous occupation states.
The information about the presence is acquired by means of a sensor of presence while the output is executed by means of a switcher.

2.4 THE IMPLEMENTED NEURAL NETWORK

The neural net is fed with the last N values of the presence parameter, that are temporally spaced according to the desired precision, and the net must predict the next value of the output that is the net must learn to predict the state of occupation of the room basing on the previous occupation states of the room.
The general scheme of the used neural net is composed by only one neuron with N inputs and a linear transfer function, as shown in the following figure.

Fig.6 Neuronal model used



The used net is able to learn using a least mean square error algorithm that is a supervised training method where a set of Q couples of vectors $[\mathbf{p_1},\mathbf{t_1}],[\mathbf{p_2},\mathbf{t_2}],....,[\mathbf{p_Q},\mathbf{t_Q}]$ are presented to the net, being **p** an input vector and **t** a target vector, and the sum of the average of the square errors between the output of the net with a given input and the desired output is calculated:

$$\frac{1}{Q}\sum_{j=1}^{Q} e_j^{\,2} = \frac{1}{Q}\sum_{k=1}^{Q}\left[t_j - a_j\right]^2 . \qquad (5)$$

The weights and the biases are adjusted to reduce the error expressed by the expression (5).
It is possible to demonstrate that for this kind of net the error has a quadratic expression and the performance index can show a global minimum, a weak minimum or no minimum, depending on the input vectors.
The least mean square error algorithm is traduced into the Widrow-Hoff learn algorithm where the weight matrix **W** and the bias vector **b** are iteratively updated according to:

$$\mathbf{W}(k+1) = \mathbf{W}(k) + 2\alpha\mathbf{e}(k)\mathbf{p}^T(k) \qquad (6a)$$

$$\mathbf{b}(k+1) = \mathbf{b}(k) + 2\alpha\mathbf{e}(k) \qquad (6b)$$

until a convergence takes place. In eqs.(6) **e** is the error vector and α is the learning rate. If α is too large learning occurs fast but if it is too large the algorithm can become unstable, diverging and increasing the error instead of reducing it. To avoid divergence the learning rate must be less than the reciprocal of the largest eigenvector of the correlation matrix $\mathbf{p}^T\mathbf{p}$ of the input vectors.
The operative scheme of the neural net is shown in figure 7.

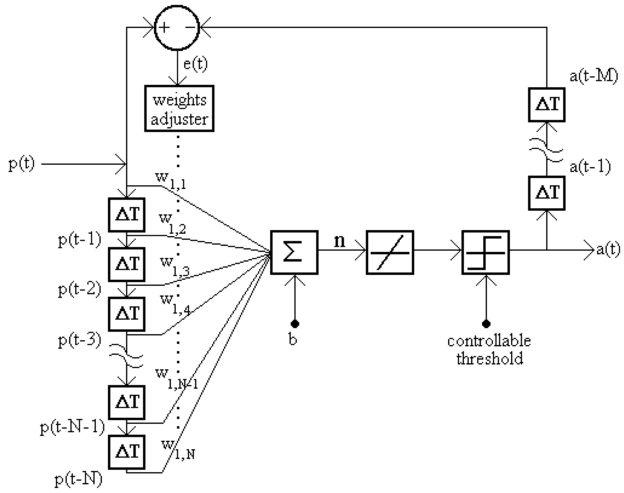


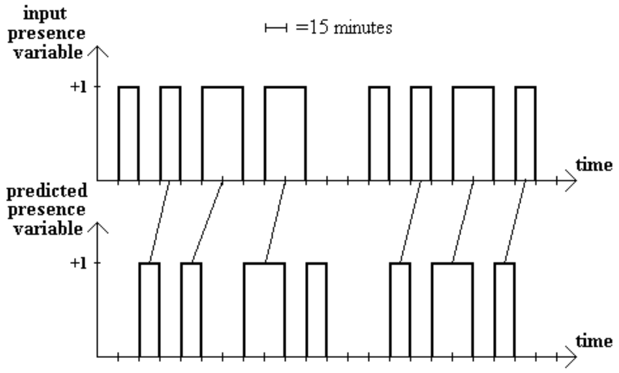
Fig.7 Operative scheme of the used neural network

The elements pointed with ΔT are delay elements that give as output their input after a time interval equal ΔT. The system acts as a predictive filter that estimates the actual value of the input variable once known N previous value of the same variable.
The actual value of the presence variable, that is a binary variable, is fed into the system that calculates the predicted value of the presence variable, necessary to decide if the system can switch the electrical loads on or off. The system uses the actual value of the presence variable and its previous N-1 values.
The prediction interval can be decided by introducing a certain number of delays elements in the back loop. In fact, since the back loop is used to calculate the error between the actual value and the predicted value, if we want the net to predict what is the value of the presence variable M·time intervals ΔT in advance, it is necessary to introduce M delays units. In this way the M-th delay unit gives at its output the presence variable predicted M time intervals in advance and we wish this variable to be equal to the actual presence variable. The comparison between these two variables is made by the error unit that executes their difference: if the error is different from zero the weight adjuster unit trains the net to improve the prediction capability of the system. If the error is equal to zero it means that the system was able to predict the actual value of the presence value M time intervals in advance and that it works well. The neural net is trained by adjusting the weights and the bias of the net using eqs.(6).
Since the presence variable is a binary variable that can assume only the values 1 or 0 and the learning function is linear, the output of the neuron can assume any value. Since the output variable has to control electrical loads, it must assume only the value 1 or 0, as the input variable, and it is necessary to introduce a controllable threshold unit, where the threshold is given as input, that generates an output equal to 1 if the input is above the threshold and an output equal to zero if the input is below the threshold. In this way it is possible to choose the characterization of the net between an energy wasting or a not energy wasting behavior. In fact if the neuron recognizes a known situation it gives as output a well defined value that can be 1 or 0. If the neuron does not recognize a known situation it gives as output a value between 0 and 1. If we prefer the system to respect the comfort of the user with respect to the energy saving we set the threshold very low so that the electrical loads are preferably set on. On the contrary if we prefer the system to neglect the comfort of the user with respect to the energy saving we set the threshold very high so that the electrical loads are preferably set off.

2.5 PRACTICAL IMPLEMENTATION

It is obvious that the more is the number of delay units, that is the number of past values of the presence variable, and the higher is the precision of the system. For energy control applications a 15 minutes delay time ΔT represents a good compromise between velocity and precision of the system.
The prediction capabilities of the system are mainly used to disconnect electrical loads that has been left switched on even if the occupant has left a given room.
It could be said that for this kind of application it is possible to use a proper delayed switcher that, after a certain time, switches the electrical load off: this does not represent a good solution for the comfort of the occupant and for the durability of the controlled load since important devices such as air conditioner would continuously switch on and off changing the inside temperature in the same way.
The used system, on the contrary, is capable of disconnecting the electrical loads learning from the behavior of the occupant that is to distinguish when it has momentarily left the controlled room from when it has definitely left the same room.
The prediction capabilities of the system are also used to switch on the electrical loads in advance with respect to the entrance time of the occupants inside the room to let them find the most comfortable situation inside the room.
To ensure a high degree of precision it is necessary to extend as more as possible the number of past input presence variables: the best results are obtained if a 24 hour period is used as input that is a total of 88 inputs (24 hour multiplied 4 samples per hour).
The used net needs a certain training time before predicting, with a high degree of precision, the behavior of the presence variable inside the controlled room that, for our system, is equal to the input period of 24 hours.
Since the standard temporal length of the presence variable has been chosen to be equal to 15 minutes, there must be a criterion that decides the binary value of the presence variable in the case of continuously changing of this variable inside the 15 minutes interval which means that the occupant enters and exits continuously. For this reason a proper threshold unit has been used. This unit acquires the instantaneous values of the presence variable and decides, at the end of the period ΔT, if the occupant has remained inside the room more than 50% of time, setting the input variable to the net equal to 1, or less than 50%, setting the input variable to net equal to zero.
The obtained results are shown in figs.8 where it is possible to see how the system make some prediction mistakes during the training time but how it correctly predicts the behavior of the user after it has worked and

learned for a certain time. The used net has been trained to



predict 15 minutes in advance.

Fig.8a Behavior of the system during the training.

Fig.8b Behavior of the system after the training.

2.6 CONCLUSIONS

We have presented a versatile system, based on neural technology, to manage and control the energy in buildings. It is able to change and adapt its management strategy to the controlled environment. The neural technique adopted is properly designed to be implemented in a low computation capabilities device such a common electronic microcontroller.

2.8 REFERENCES

[1] S.Mc Clelland, "Intelligent building", New York: Springer-Verlag, 1989.

[2] J.A.Bernaden, R.E.Neubauer, "The intelligent building sourcebook", Lilburn: Fairmont Press, 1988.

[3] V. Bradshaw, K. E. Miller ,"Building Control Systems", John Wiley & Sons.

[4] V. Boed, "Networking and Integration of Facilities Automation Systems", CRC Press.

[5] M. Eyke,"Building Automation Systems: a Practical Guide to Selection and Implementation", Blackwell Science (UK).

[6] D. A. Wacker, "Complete Guide to Home Automation", Popular Woodworking.

[7] B. Gerhart, "Home Automation Guide for Builders", McGraw-Hill Publishing.

[8] F. Garzia, G. Veca, “Smart Automatic Control of Energy Flows in Building”, DUE 2000, Cape Town.

[9] N. K. Bose, P. Liang, “Neural Network Fundamentals with Graphs, Algorithms, and Applications”, McGraw-Hill Publishing Company, 1998.

[10] S. I. Gallant, “Neural Network Learning and Expert Systems”, MIT Press, 1997.

[11] W. T. Miller, “ Neural Networks for Control (Neural Network Modeling and Connectionism), MIT Press, 1999.

2.9 AUTHORS

Principal Author: Fabio Garzia holds a degree in Electronics Engineering and a PhD in Applied Electromagnetism from the University of Rome "La Sapienza", Italy. He is actually lecturers of specialized modules of security and automation subjects in different courses at the Faculty of Engineering of the same University. He is founder and responsible, together with Prof. Mario Bertolotti, of the Applied Security Laboratory (www.securitylab.org) at the Department of Energetics of the University of Rome "La Sapienza", Italy. He is also the designer of security, supervision and automation installations for the Italian State Railways at Italferr SpA - FS Group. His address is:

Dipartimento di Energetica
Università degli Studi di Roma "La Sapienza"
Via A. Scarpa, 16
00161 Roma, Italy

Co-author: Giuseppe M. Veca is full professor of Electrical Engineering at the University of Rome "La Sapienza", Italy. He is Senior Member of IEEE and Dean of the degree in Electrical Engineering. His fields of research are EMC/EMI, Magnetic Devices and Power Apparatus and Systems. His mail address is:

Dipartimento di Ingegneria Elettrica
Università degli Studi di Roma "La Sapienza"
Via Eudossiana, 18
00184 Roma, Italy

Presenter: The paper is presented by Fabio Garzia