Clima p181

AIVC 10936

NATURAL VENTILATION AND ARTIFICIAL NEURAL NETWORKS

KINDANGEN J.I.*°, KRAUSS G.* and DEPECKER P.*

*CETHIL/ ETB, INSA de Lyon, 20 Avenue Albert Einstein, Bâtiment 307, 69621 Villeurbanne Cedex, FRANCE
°University of Sam Ratulangi, Faculty of Engineering, Architectural Dept., Manado 95115, INDONESIA

Abstract

This study presents a new method of interior air motion assessment using artificial neural networks. The air motion inside a building depends not only on the external wind velocity, but also to a great extent on most of architectural parameters such as position and orientation of building, size and configuration of windows, roof geometry, whether the building is stilted or not, etc.. The difficulty to evaluate the interior velocity coefficient, a non-dimensional parameter that is the measure of relative strength of the interior air movement, if we would take into account a number of architectural parameters; this encouraged us to use this approach.

After presenting the general setting of our work, we introduce the neural networks in describing their main properties and the methods of their implementation. We have applied these ideas to our study and presented the initial obtained results.

The utilization of the neural networks as a *model-free* predictor is a way of interesting investigation which facilitates designers or architects to take into account a number of influential parameters in natural ventilation investigating. Moreover, this allows to assess indoor airflow pattern without doing a costly experiment or running an expensive and complicated flow field simulation code.

Keywords: architectural parameters, artificial neural networks, interior velocity coefficient, naturally ventilated buildings, humid tropical climate

1. INTRODUCTION

Air motion is one of the primary factors which determine human comfort in hot climate, especially in humid regions. When a sufficient airflow to permit a good ventilation exists, we can obtain comfort in buildings without using any systems of active air-conditioning. In association with other devices such as efficient solar protections and good insulation of partitions, natural ventilation maintains thermal comfort in buildings. Air motion acts on thermal comfort, decreasing the risks of local overheating and permitting sufficient interior air velocity to improve the evaporation of sweat. The air motion inside a building depends not only on the external wind velocity, but also largely on a number of architectural parameters.

The interior air flow behavior can be analyzed using methods such as measurement *in situ*, experimental study in wind tunnels and numerical simulation with computational fluid dynamics (CFD) codes. However, these approaches remain generally too expensive and difficult as a design tool at present for the majority of the designers or the architects [1-3].

Chen et al [3] has established the database for assessing indoor airflow pattern, he also proposed that since the database will only be able to store a limited number of the cases with several different parameters, different levels of interpolation should be outlined to interpolate the results from such a database. "Interpolation" within a database is the art of deducing useful information for a specific design from a database. However, this work has not covered the problem of air flow through large openings which is the major element of tropical architectures.

Moreover, there are many possibilities of architectural parameters effect on interior air flow, so it is hard work to investigate the effect of all possibilities if we shall use one or all of the three investigations described above.Thus, it is necessary to develop a predicted method as a design tool. An architect or designer should be able to use the tool without using turbulence modelling knowledge or performing a cost experiment. The results obtained by the tool should also be realistic and accurate. With the help of these two latter approaches we established a database concerning the influence of architectural design elements on interior airflow; with which we afford the training phase of the artificial neural nets (ANN).

ANN can estimate input-output functions, unlike statistical estimators, they estimate a function without a mathematical model of how output depends on input. They are *model-free* estimators and *learn from experience* with numerical sample data. We can program or train neural networks to store, recognize and retrieve patterns or database entries; to solve combinatorial optimization problems; to filter noise from measurement data; to control ill-defined problems; in summary to estimate sampled function when we do not know the form of the functions [4].

This paper presents the prediction of architectural parameter effects on interior air motion using neural networks. After presenting the general setting of our work, we introduce the neural networks in describing their main properties and the methods of their implementation. We have applied these ideas to our study and presented the first obtained results.

2. INTERIOR VELOCITY COEFFICIENT

The complete determination of velocity field in the building requires either the utilization of experimental investigation or the utilization of numerical simulations (CFD codes). The measuring of the relative strength of the interior air movement in the horizontal plane, which is representative of the occupied space of the room generally uses a non-dimensional parameter called the coefficient of velocity.

The velocity coefficient was used that is defined as the ratio of the mean air velocity in the zone studied with the air velocity as a reference point in the exterior flow at windward of building.

$$C_V = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{V_i}{V_r}\right) \quad (1)$$

Where, C_v is coefficient of velocity, V_i is mean velocity at interior location i (ms^{-1}), V_r is mean outdoor reference free-stream velocity at the reference height (ms^{-1}), and n is number of points measured.

The simple formalism of this coefficient definition conceals the difficulty of its determination. This determination is carried out by one of the following:

- *In situ* experimentation on cells or buildings [5]. This method is not usable in the design phase, except if the project involves existing structures.
- For some models wind tunnel experimentation [6-8]. This method facilitates the study of architectural parameters, but remains difficult.
- Numerical simulation (computational fluid dynamics) [9-13]. This method becomes easy to handle by the conjoined improvement of some computers' performances and some algorithms of resolution which are more accessible.

In practice, these ways of investigating constitute a database of usable knowledge in order to work out some rules of design or in order to define a correlation. Several authors [7, 8] proposed thus the correlation used to evaluate the interior air motion affected by some architectural parameters. Generally the proposed correlation carries on some simple profiles and takes into consideration one or two influential parameters.

To take into account the multiple parameters (> 3) in the establishment of the correlation is very difficult if not impossible. This encouraged us to try a new way, using neural networks.

3. ARTIFICIAL NEURAL NETWORKS

Artificial neural networks were invented in the spirit of a biological metaphor. The biological metaphor for neural networks is the human brain. Like the brain, this computing model consists of many small units that are interconnected. These units (or nodes) have very simple abilities. Hence, the power of the model is derived from the interplay of these units. It depends on the structure of their connections.

In 1943, MacCulloch and Pitts proposed a mathematical pattern of biological neuron [14]. This type of neuron possesses a very simple dynamic. The input signal is valued as the weighted sum of excitations coming from the outside. From a mathematical point of view, a feed-forward neural network is a function. It takes an input and produces an output. The input and output are represented by real number. A simple neural network may be illustrated like in Fig. 1.

This network consists of seven units or neurons or nodes (the circles) and twelve connections (the arrows). The terms w_{ij} applied to each connection are called weight, it indicates the strength of the connection. Connections with a positive weight are called excitatory, the ones with a negative weight are called inhibitory. The constellation of neurons and connection is called the architecture or the topology of the network.

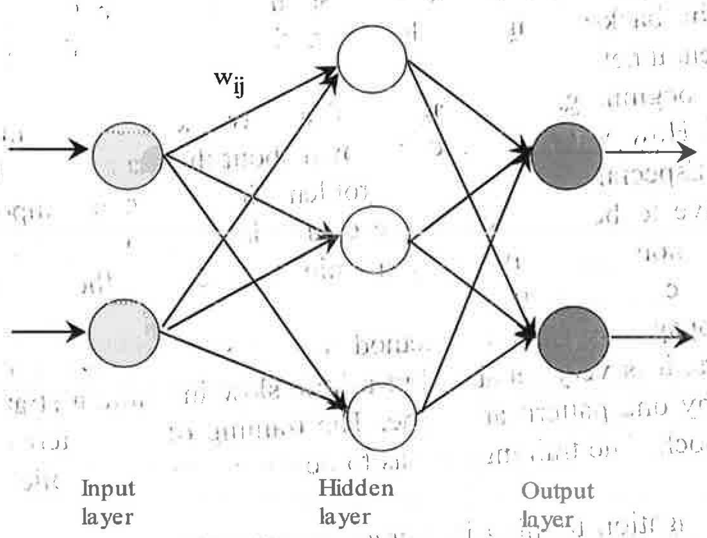


Fig. 1. A neural network.

Fig. 2 illustrates how information is processed through a single node. The node receives the weighted activation of other nodes through its incoming connections. First, these are added up (summation). The result is passed through an activation function, the outcome is the activation of the node. For each of the outgoing connections, this activation value is multiplied with the specific weight and transferred to the next node.

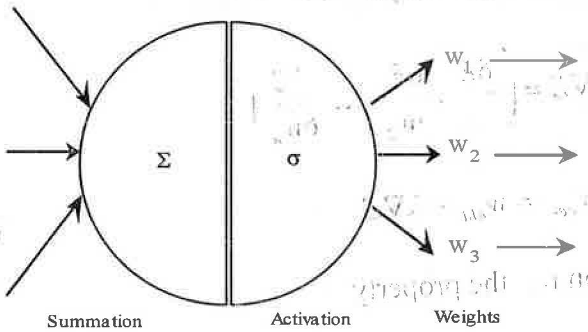


Fig. 2. Information processing in a neural network unit.

A few different threshold functions are used. It is important that a threshold function is non-linear, otherwise a multilayer network is equivalent to a one layer net.

The most widely applied threshold function is the logistic sigmoid like the following formula:

$$\sigma(x) = \frac{1}{1+\exp(-x)} \qquad (2)$$

There are a few other activation functions in use: scaled sigmoid, gaussian, sine, hyperbolic tangent, etc. However, the sigmoid is the most common one. It has some benefits for back-propagation learning, the classical training algorithm for feed-forward neural networks.

At the beginning, the weights of a network are randomly set or otherwise predefined. However, only little is known about the mathematical properties of neural networks. Especially, for a given problem, it is basically impossible to say which weights have to be assigned to the connections to solve the problem. Since ANN follow the non-declarative programming paradigm, the network is trained by examples, so called patterns.

Back-propagation is one method used to train the network. It has many predecessors, it is very reliable, but a little slow in training strategy. The training is performed by one pattern at a time. The training of all patterns of a training set is called an epoch. The training set has to be a representative collection of input-output examples.

Back-propagation training is a gradient descent algorithm. It tries to improve the performance of the neural net by reducing the error along its gradient. The error is expressed by the root-mean-square error (RMS), which can be calculated by:

$$E = \tfrac{1}{2}\sum_{p} \left\| t_p - o_p \right\|^2 \qquad (3)$$

The error (E) is the half the sum of the geometric averages of the difference between projected target (t) and the actual output (o) vectors over all pattern (p). In each training step, the weights (w) are adjusted towards the direction of maximum decrease, scaled by some learning rate lambda (λ).

$$\nabla E = \left(\frac{\delta E}{\delta w_1}, \frac{\delta E}{\delta w_2}, \ldots, \frac{\delta E}{\delta w_n} \right) \qquad (4)$$

$$w_{new} = w_{old} - \lambda \nabla E \qquad (5)$$

The sigmoid function has the property

$$\frac{d}{dx}\sigma(x) = \sigma(x)(1-\sigma(x)) \qquad (6)$$

Thus the derivative of the sigmoid can be computed by applying simple multiplication and subtraction operators on the results of the sigmoid function itself.

This simplifies the computational effort for the back-propagation algorithm. In fact, the equations for weight changes are reduced to:

$$\Delta w_{from,to} = -\lambda o_{from} \delta_{to} \qquad (7)$$

$$\delta_{output} = -(t_{output} - o_{output}) \qquad (8)$$

$$\delta_{hidden} = \sigma'(s_{hidden}) \sum_{i} \delta_{i} w_{hidden,i} \qquad (9)$$

There are different functions for connections to hidden and output nodes. The unprocessed sum (s) for each neuron has to be stored before the activation function is applied to it. Then, basic algebra operations like multiplication and subtraction are sufficient to perform the weight changes.

4. APPLICATION OF THE NEURAL NETWORKS

A neural network functions as a universal predictor. It can learn a pattern set and output as it would dictate on unknown patterns through generalization over the pattern set. Here, it will be applied to the prediction of interior velocity coefficients as a function of architectural parameters (PA_N). A C_v can be formulated as follows:

$$C_v = f\,(\,PA_N\,) \qquad (10)$$

The database provides information about average interior air velocity presented by interior velocity coefficient due to any architectural elements. The structure of the database is as follows. It consists of a number of limited pairs: a combination of architectural parameter and its interior velocity coefficient. Since the effect of a new combination of architectural parameter on interior airflow pattern will be predicted, we could use ANN for this purpose.

Among the elements which should be taken into consideration in the design of a network are the following:

- The constitution of a training set and a generalization set.
- The precise architecture of the network.
- The choice of training parameters.

There are no reliable guidelines for determining the appropriate number of neurons in a hidden layer, or for deciding how many hidden layers there should be [15]. Stevenson [16] has shown that two hidden layers are enough to produce acceptable results. Networks with more than two hidden layers are rare, mainly because they are difficult to train. The constitution of the training set involves obtaining the input and the output parameters. In the present case, the inputs are the wind directions and some architectural parameters; the outputs are the interior velocity coefficients. The larger the space of the training set and the width of the input spectrum it covers, the better the results achieved by training.

Table 1 shows the matrix used for the input data and the associated C_v values. It contains the combination of architectural parameters treated: the orientation of the building, the form and the configuration of the roof, whether the building is stilted or not, floor penetration, overhangs, the size and location of windows and the presence of wind cheeks or mosquito-nets on any of the four sides. In the present case, two types of training set were used. The first incorporated the real values of the input parameters and the C_v. For the second, the numerical values were normalized by the formula:

$$x_i = \frac{(x - x_{min})}{(x_{max} - x_{min})} \tag{11}$$

where, x_i is normalized value, x is original value, x_{min} is minimal original value, and x_{max} is maximal original value.

Normalizing of the training inputs generally improves the quality of the training. The training was based on results obtained by numerical simulation (*STAR-CD* and *FLUENT*) [9-13], and on measurements made in a numerical wind tunnel [6]. Fig. 3 illustrates architectural parameters treated.

INPUTS:

Architectural parameters / Number of cases	1	2	...	n	Encoding	Normalized encoding	
Building orientation		45			[0,330]	0.136	
Arc of longitudinal section		1			[0,1]	1	
Angle of longitudinal section 1		0			[0,90]	0	
Angle of longitudinal section 2		0			[0,90]	0	
Angle of transverse section 1		90			[0,90]	1	
Angle of transverse section 2		90			[0,90]	1	
Arc of transverse section		0			[0,1]	0	
Stilts		1			[0,1]	1	
Floor penetration		0			[0,1]	0	
Number of windows on windward side		2			[0,2]	1	
Porosity of windward wall		30			[0,100]	0.30	
Wind cheeks on windward side		2			[0,4]	0.50	
Overhangs		1.2			[0,2]	0.60	
Windows with mosquito-nets		0			[0,2]	0	
Number of windows on leeward side		2			[0,2]	1	
Porosity of leeward wall		30			[0,100]	0.30	
Wind cheeks on leeward side		2			[0,4]	0.50	[0,1]
Overhangs		1.2			[0,2]	0.60	
Windows with mosquito-nets		0			[0,2]	0	
Number of windows on right side		0			[0,2]	0	

Porosity of right-side wall		0			[0,100]	0	
Overhangs		1.2			[0,2]	0.60	
Windows with mosquito-nets		0			[0,2]	0	
Number of windows on left side		0			[0,2]	0	
Porosity of left-side wall		0			[0,100]	0	
Overhangs		1.2			[0,2]	0.60	
Windows with mosquito-nets		0			[0,2]	0	

OUTPUT:

Interior velocity coefficient (Cv)	Cv_i	0.57	...	Cv_n	[0,1]	0.57	[0,1]

Table 1. A matrix of training and generalization.

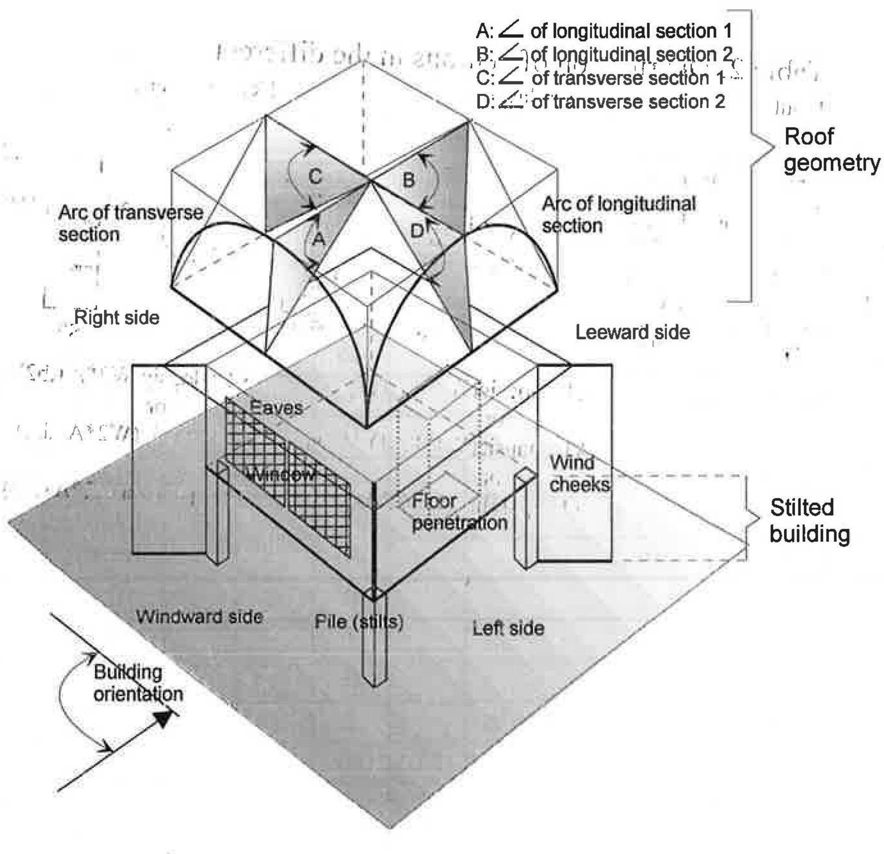


Fig. 3. Architectural parameters.

The evaluation of the performance of the networks was based on the distribution of the inputs used in the numerical simulations (CFD) of 30 models, involving 97 cases, and by wind tunnel tests of 15 models, involving 119 cases:

- The data were divided into two groups, one for training and the other for generalization or validation. The training group had 166 cases, and the generalization group had 50 cases.

For the architecture of the network (Fig. 4), the following structures were studied:

a. Networks with 3 layers: 1 input layer - 1 hidden layer - 1 output layer.

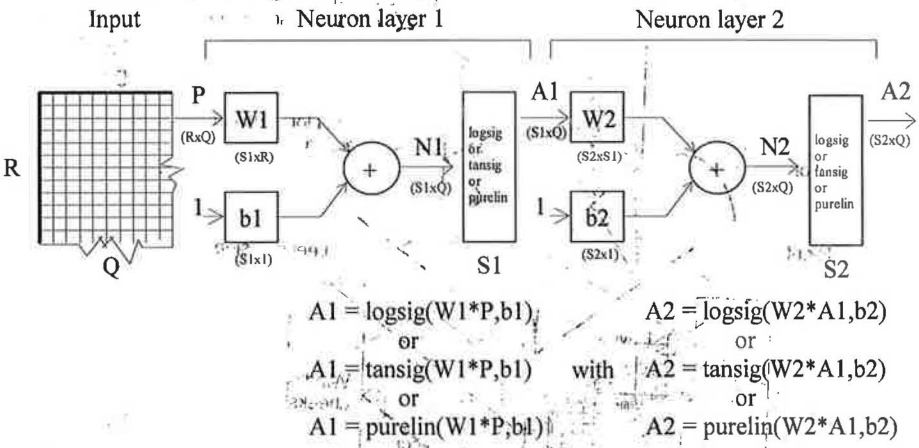
b. Networks with 4 layers: 1 input layer - 2 hidden layers - 1 output layer.

The following activation functions were used:- sigmoid logistic function (*logsig*), hyperbolic tangent function (*tansig*) and linear one (*purelin*).

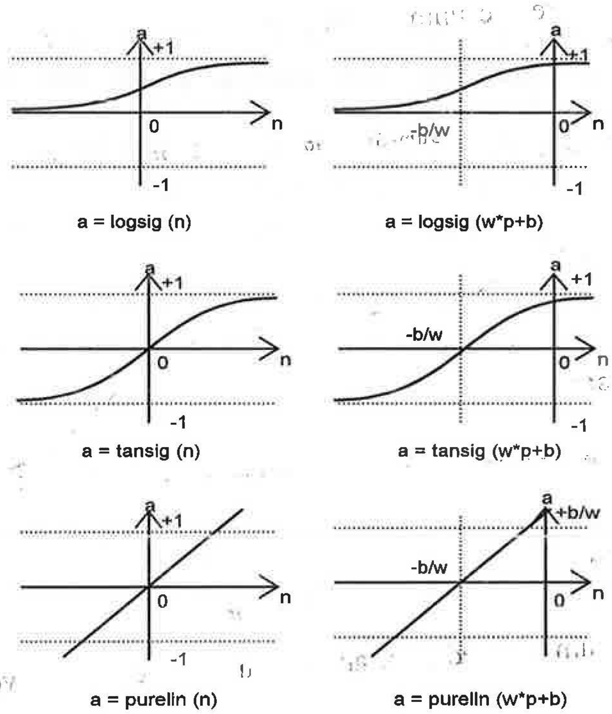
The number of neurons in a given layer can vary. In the present study they were as follows:

	ANN with 1 hidden layer	ANN with 2 hidden layers
Number of neurons (S1)	5, 10, 15, 20, 30, 50, 100, 500	5, 10, 14, 20, 28, 50
Number of neurons (S2)	1	5, 10, 14, 20, 28, 50
Number of neurons (S3)	-	1

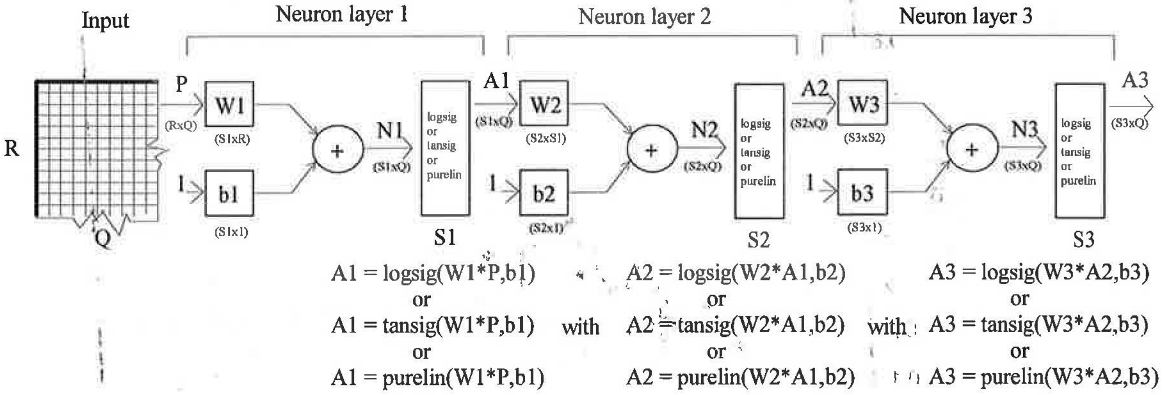
Table 2. Distribution of neurons in the different layers.



(a)



(b)



(c)

Fig. 4. An architecture of neural networks with 1 hidden layer (a), some used transfer functions (b) and neural networks with 2 hidden layers (c).

The types of training used were back-propagation, back-propagation with momentum, and Levenberg-Marquardt optimization. The training algorithms have been implemented in the software package *MATLAB* [17].

5. RESULTS AND DISCUSSION

A number of networks with different numbers of hidden neurons were run (133 networks with one hidden layer, 112 networks with 2 hidden layers), and the network with the lowest validation error (the minimum mean error in generalization) was taken as the optimum configuration. Initially, the errors in both the training set and the generalization set should fall; however, if the generalization error starts to rise, this means that there are unmodeled dynamics in the net. Further training will then be counterproductive.

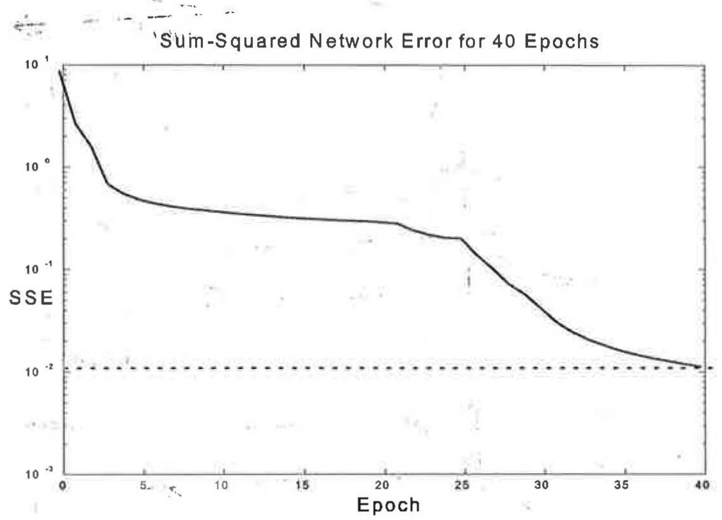


Fig. 5a. Sum squared error and epochs for the best network's configuration.

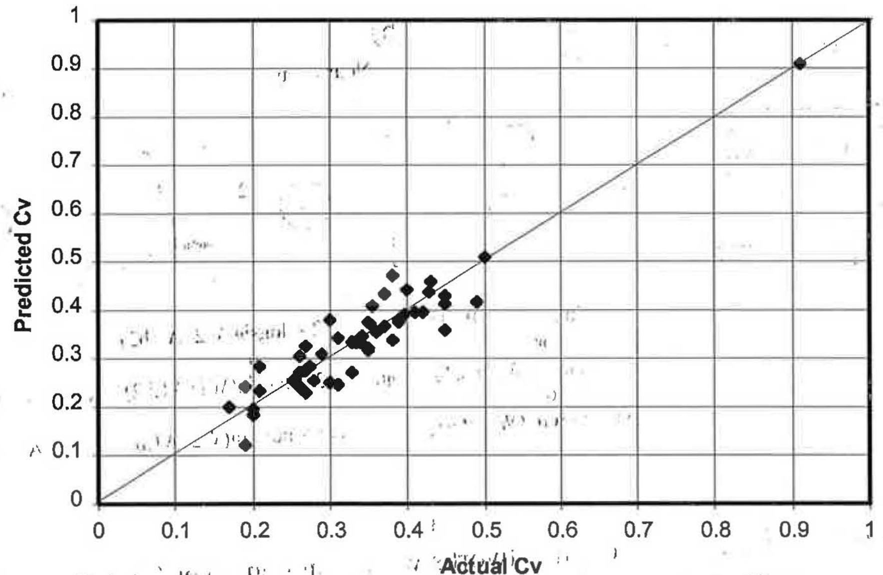


Fig. 5b. Actual C_v versus predicted C_v in generalization for the best one, $R^2 = 0.882$ and SD = 0.115.

The results given in Fig. 5 were obtained by Levenberg-Marquardt back-propagation with 2 hidden layers containing 20 neurons each, the transfer function being a hyperbolic tangent function for all the neurons. These networks provided reliable results: 3.11% of mean error in generalization. The best results obtained are shown in the Table 3, along with the network configurations used.

Best results obtained		
	ANN with 1 hidden layer	ANN with 2 hidden layers
Mean error in training	0.43%	0.46%

Mean error in generalization	5.1%	3.11%
No. of neurons (S1), activation f.	10, tansig	20, tansig
No. of neurons (S2), activation f.	1, logsig	20, tansig
No. of neurons (S3), activation f.	-, -	1, logsig
No. of epochs	15 epochs	40 epochs
Back-propagation method	Levenberg-Marquardt	Levenberg-Marquardt
Error goal	0.01	0.01
Type of training set	normalized	normalized

Table 3: The best results obtained.

The choice of the inputs to the training set is crucial to the quality of the results. And the more space there is in the training set, the better the predictions. It is indispensable to ensure that the cases included in the training set cover the whole range of variations of the parameters.

In spite of the proposed approach covers the limited type of prediction: in single-zone and output -- the interior velocity coefficient, we show how this approach can be applied at present and will be expanded as a design tool in the future.

A limited database has been stored and used to train the neural nets. The nets keep their final weights that are ready to use in the prediction. To outline the average indoor velocity as a function of a new combination of architectural parameters which was not presented in the database or the training set, it is enough to entry their numerical values. For each parameter, this varies between the minimal and maximal value as they were indicated in Table 1. For example, it can predict the effects of a building orientation variable that varies between 0-330° clockwise from North. The procedure of prediction is really identical with the generalization or validation one.

The principle of this calculation is practically resided in their final weights, this is important to note that if the ANN have been trained, their final weights are as a usable form to commence the prediction process (Fig. 6). In our recent work, we have integrated this approach in the modular programming using TRNSYS [18] to evaluate the thermal comfort in the humid tropical climate. Obviously, the algorithms give the reliable and satisfied model and can be linked very well with the other module.

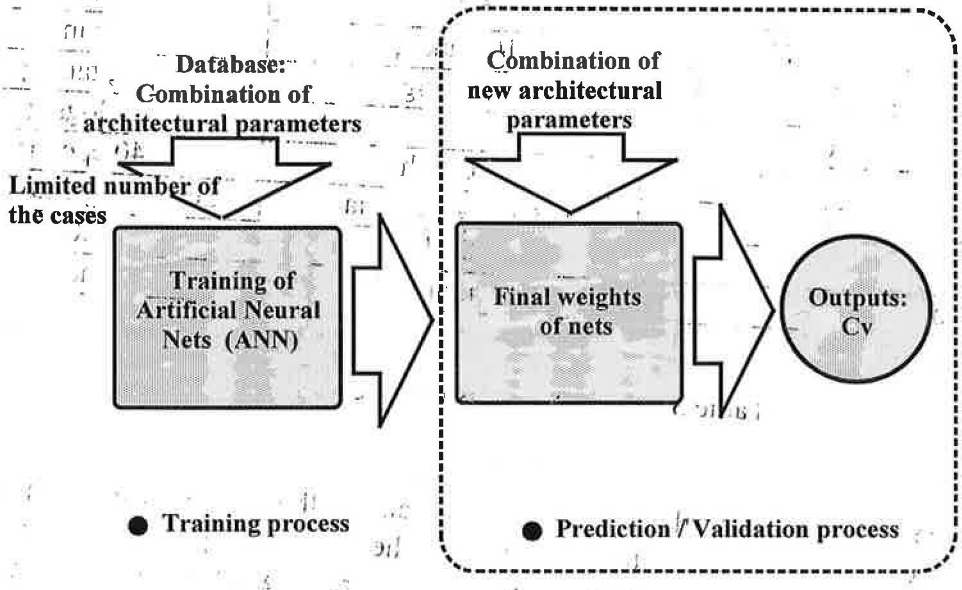


Fig. 6. ANN's applicability to evaluate indoor airflow due to a combination of architectural parameters.

In order to serve this method as a design tool in the future, it is interested to cover more and more parameters such as multi-zone airflow problems. The interest of this method holds in the fact that it is relatively easy to integrate numerous parameters. It is also important to establish the viability of this approach to predict the others' coefficients which are usable in evaluation of indoor airflow pattern such as the spatial variation, maximum velocity, local indoor velocity, average pressure coefficients, etc. In the case of expanding the number of cases, of parameters and of output which will be treated, it was sufficient to restart the phase of network training in order to adapt the weight. Fig. 7 shows a scenario of the further works to develop this method as a design tool that will cover more parameters.

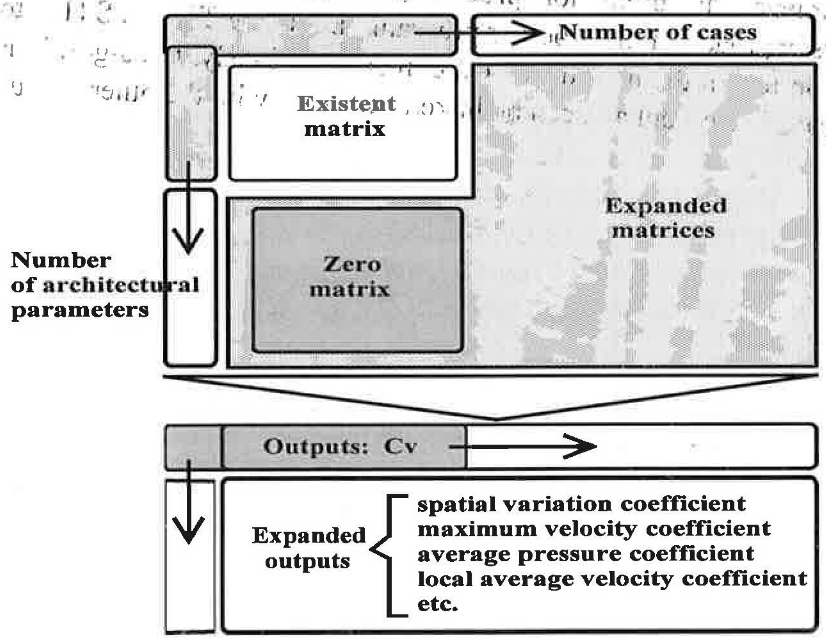


Fig. 7. A scenario of the further development as a design tool.

6. CONCLUSIONS

The enlargement of the training set requires only a re-actualization of the obtained weights by a new phase of training, and not a complete redefinition of the network. A certain number of disadvantages should, however, be mentioned:

- The knowledge contained in a network is not legible, unlike that which is contained in the rules of an expert system.
- In the case presented here, the constitution of a training set is a difficult task, requiring the implementation of other means (experimental or numerical).
- Network design is still a highly empirical activity, for which no general rules exist as yet.

In sum, the use of neural networks as universal predictors opens up an interesting field of investigation. They provide reliable results in cases where a number of parameters have to be taken into account simultaneously, and thus make it possible to memorize applied measurements or numerical simulations in usable form.

7. REFERENCES

1. Nielsen, P.V., Progress and Trends in Air Infiltration and Ventilation Research. In *Proceedings of the 10th AIVC Conference*, Air Infiltration and Ventilation Centre, Dipoli (1989)
2. Jones, P.J. and Whittle, G.E., Computational Fluid Dynamics for Building Air Flow Prediction -- Current Status and Capabilities, *Building and Environment*, vol. **27** no.**3**, pp. 321-338 (1992)
3. Chen, Q., Moser, A. and Suter, P., *A Database for Assessing Indoor Airflow, Air Quality, and Draught Risk*, International Energy Agency, Annex 20, Subtask 1 (1992)
4. Kosko, B., *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*, Prentice-Hall International Editions, Englewood Cliffs, NJ (1992)
5. Celestine, C., *Ventilation naturelle et confort thermique dans l'habitat en climat tropical humide - Approche expérimentale en vraie grandeur de la ventilation naturelle sur un site de la Guadeloupe - Etude comparative en soufflerie*, Thèse Docteur Ingénieur, INSA de Lyon (1985)
6. Gouin, G., *Contribution Aérodynamique à l'Etude de la Ventilation Naturelle de l'Habitat en Climat Tropical Humide*, Thèse de Docteur, Université de Nantes (1984)
7. Ernest, D. R., Bauman, F. S., Arens, E. A., The Prediction of Indoor Air Motion for Occupant Cooling in Naturally Ventilated Building, *ASHRAE Transactions*, vol. **97** part **1**, pp. 539-552 (1991)
8. Givoni, B., *L'Homme, L'Architecture et Le Climat*, Editions du Moniteur, Paris (1978)
9. Kindangen, J. I., *Modélisation Aéraulique en Climat Tropical Humide, d'un Bâtiment Largement Ouvert sur l'Extérieur*, mémoire de DEA, CETHIL/EEH, INSA de Lyon (1994)

10. Kindangen, J. I., Krauss, G., Investigation of Natural Ventilation with Computational Fluid Dynamics. A Comparison Study with Wind Tunnel Results, *Architectural Science Review*, vol. **39** no. **2**, pp. 113-120 (1996)
11. Kindangen, J. I., Krauss, G., Depecker, P., Effects of Roof Shapes on Wind-Induced Air Motion Inside Buildings, *Building and Environment*, vol. **32** no. **1**, pp. 1-11 (1997)
12. Kindangen, J. I., Krauss, G., Rousauën, G., Influence of Architectural Parameters on Air Flow Distribution Inside a Naturally Ventilated Single-Unit House, *Proc. of the 13th International PLEA Conference*, Louvain-la-Neuve, Belgium, pp. 171-176 (1996)
13. Krauss, G., Depecker, P., Kindangen, J. I., Ventilation Naturelle en Climat Tropical Humide, Proposition d'Une Méthode d'Evaluation des Coefficients de Vitesse, *Journée thématique de Société Française des Thermiciens*, Paris (1996)
14. Jodouin, J. F., *Les Réseaux Neurones, principes et définitions*, Hermès, Paris (1994)
15. Jodouin, J. F., *Les Réseaux Neuromimétiques*, Hermès, Paris (1994)
16. Stevenson, J., Using Artificial Neural Nets to Predict Building Energy Parameters, *ASHRAE Transactions*, vol. **100** part **2**, pp. 1081-1087 (1994)
17. Demuth, H., Beale, M., *Neural Network Toolbox, for use with MATLAB*, The Math Works Inc. Massachusetts (1994)
18. TRNSYS 14.1, Solar Energy Laboratory, University of Wisconsin -- Madison (1994)