Modelling non Measurable Processes by Neural Networks: Forecasting Underground Flow Case Study of the Céze Basin (Gard - France)

A. Johannet, P.A. Ayral, B. Vayssade

Modelling non Measurable Processes by Neural Networks: Forecasting Underground Flow Case Study of the *Cèze* Basin (*Gard* - *France*)

A. Johannet*, P.A. Ayral**, B. Vayssade*
Ecole des Mines d'Alès
*Centre des matériaux de Grande Diffusion
**Laboratoire de Génie de l'Environnement Industriel
6 avenue de Clavières / 30319 Alès Cedex / FRANCE

Abstract: After a presentation of the nonlinear properties of neural networks, their applications to hydrology are described. A neural predictor is satisfactorily used to estimate a flood peak. The main contribution of the paper concerns an original method for visualising a hidden underground flow Satisfactory experimental results were obtained that fitted well with the knowledge of local hydrogeology, opening up an interesting avenue for modelling using neural networks.

I INTRODUCTION

During the last twenty years there has been considerable research devoted, on the one hand, to the field of nonlinear and adaptive modelling, and on the other hand to the study of neural networks in order to perform such tasks. Nevertheless, the idea of using neural networks' ability to model nonlinear and non-stationary behaviours in hydrological systems emerged only about ten years ago. Currently, several theoretical results and many different learning schemes have proven that neural networks are becoming a very effective tool in hydrological applications.

The *Gard* Region, in the South-East of France, has a specific geographical position which makes it particularly vulnerable to flash floods: each autumn, storms formed over the sea and pushed by southerly winds provoke extreme rainfall events. Flash floods are very important natural hazards for the Mediterranean Region. Unfortunately, scientific knowledge about them is insufficient.

In that context, this paper has two objectives: the first is to present how and why the neural methods are appropriate for solving such environmental problems. The second objective of this paper is to present how the neural *black box* can be changed into a *grey box* in order increase our knowledge.

The paper contains four parts: part one introduces neural networks including nonlinear properties. The second part is devoted to presenting the principal neural architecture and learning rules. The third part presents the problematic and the fourth the results.

II NEURAL NETWORKS FOR IDENTIFICATION

System identification is the modelling of systems. It is useful for the knowledge it gives about the system and in that it provides ways to control it, and to predict or forecast its behaviour.

Neural networks are devices capable of learning. In the case of signal processing, or system identification, the set of examples consists of sampled input and output signals. The second fundamental property of neural networks is that they can implement non linear functions. This property is a necessary one for systems such as catchment areas which may have different responses even when the input is the same (for example, the behaviour during summer or winter is very different).

The Model of Neuron and Multilayer Network

An artificial neuron is a mathematical operator which generally computes two actions: first the linear weighted sum of its inputs, and second the non-linear evaluation of its output. Various models of neurons have been proposed depending on the evaluation function. The formula is:

$$o_l = f\left(\sum_{example\_m} c_{lm}.i_m\right)$$

where o_l is the output of the neuron l, i_m is one of its inputs, c_{lm} is the synaptic coefficient linking this input to the neuron under consideration, and f(.) is the evaluation function. For example it is possible to choose f(.) = tanh(.). Linear neurons may exist, they have an Identity function.

A neural network is a set of interconnected neurons. These connections (defined by the set of coefficients c_{lm}) are computed during the learning phase.

It has been demonstrated that any non linear, smooth function can be identified by such a network [1]. The accuracy of the identification depends on the number of hidden neurons. This result is of course very important, but it only constitutes a proof of the existence of the solution; therefore the difficulty is to find the solution using the appropriate learning method operating on an architecture which includes a sufficient number of neurons. In this study we firstly consider the well known two-layer perceptron, and secondly an *ad hoc* network, coding in its architecture the function we want to implement.

Learning

The neural network learning phase is the computation of the synaptic weights in order to minimise a *"goal function"*. Different learning rules can be derived taking into account different *goal functions* and different minimising methods. Let us consider only identification and forecasting applications; principally two types of *goal functions* have been proposed: supervised or unsupervised one. Amongst the unsupervised methods, the Reward-Penalty learning algorithm [2] is very interesting because it enables interpretation as the gradient descent of a *goal function* (redo or undo past action) and does not need a comprehensive modelling of the environment. We have implemented this method for solving a robotic task. The aim was to make a hexapod robot learn gait and obstacle avoidance. Results obtained both in simulations and with the real plant [3] were very satisfactory; the robot learnt its task without explicit modelling of the actuator-environment relations. This application highlights the fact that neural networks find their field of excellence when they are applied to model the real world or natural environment.

On the other hand, the cost function G is more understandable in the case of supervised learning, since this function is generally the sum of the squared errors between the measured outputs and the computed values, for each input-output couple of interest. It is possible to consider this "cost" function J as follows (only one output neuron):

$$J(C,k) = \frac{1}{2}\sum_{\{k\}}\left(o^k(C) - d^k\right)^2$$

where {k} is the set of input-output couples taken for k past values, and C is the set of synaptic coefficients.

Starting from this *cost function*, several learning rules have been proposed depending on the chosen minimising method. The most popular method has been the backpropagation learning rule introduced by D. Rumelhart [4] which uses the steepest gradient descent. However, other more efficient rules have been proposed, for example a descent inspired by second order minimisation methods [5] [6]. Amongst these second order methods the "Levenberg-Marquardt" learning rule [7] is at present the most powerful and leads in a few iterations to a very satisfactory solution.

Backpropagation learning rule

The backpropagation learning rule provides a method for modifying the network's synaptic weights according to the gradient of the quadratic error. It was the first learning rule which enabled learning on nonlinear networks, and which could also efficiently operate on multilayer networks.

Let us consider the network shown in Figure 3. An input-output couple is presented to the network which has to associate the input vector i^k $\{i^k_1, i^k_2, i^k_3, \ldots\}$ to the desired output d^k (scalar value in case of one output neuron). It can be noticed that the intermediate, or hidden, neurons have no desired value. After computation of the network's output o^k, the modification to apply to the coefficients, at time t, using a gradient method with a constant step μ is:

$$c^k_m(t+1) = c^k_m(t) - \mu\frac{\partial J(C;t)}{\partial c^k{}_m} \qquad (4)$$

Therefore, using the backpropagation learning rule, the synaptic coefficients of a multilayered neural network can be computed. Its principal drawbacks are the sensitivity of the result to the initialisation of the synaptic weights, and the slowness of the convergence rate toward a minimum of the *cost function*.

Levenberg-Marquardt Learning Rule

Because of its efficiency, the Levenberg-Marquardt rule should be used whenever possible. Nevertheless, the Levenberg-Marquardt learning rule suffers from two drawbacks: first it has to invert a matrix which is an approximation of the Hessian: the second order derivative of the *cost function* relative to the synaptic coefficients, *i.e* a matrix whose dimension is equal to nc.nc if nc is the number of synaptic coefficients. Sometime this matrix is too huge to be inverted; sometimes this Hessian matrix may be non-invertible [8]. We will see later that in case of hydogeological modelling, the data are very noisy and lead to difficult problems for which Levenberg–Marquardt algorithm may be inefficient. In such cases, the backpropagation algorithm provides adequate results.

In some words (see [7][8] for full presentation), Levenberg-Marquardt algorithm starts, as bakpropagation, from a problem of cost function minimization. The principle of the rule is to apply to the coefficients an increment taking into account the first and second order of the Taylor decomposition of the *cost function* (notes that Levenberg-Marquardt addresses the *cost function*, taking into account the whole set of learning couple at the same time t). Noting that the second term of the Taylor decomposition needs the computation of the Hessian Matrix, Levenberg–Marquardt method considers an approximation of the Hessian:

$H=\Delta^T\Delta$, where Δ is the vector composed of the first order derivative of the *cost function* (computed by the backpropagation), the formula is:

$$[\mathrm{H}]_{ij,lm} \cong \sum_{\{k\}}\frac{\partial J}{\partial c_{ij}}\frac{\partial J}{\partial c_{lm}}$$

The Levenberg-Marquardt rule assumes that at each presentation t of the whole set of learning couples $\{i^k,o^k\}$, an increment to the coefficients is computed in the direction of the gradient: Δ, with amplitude μ(C,t) such that:

$$\mu(\mathrm{C},t) = \left(\Delta^T\Delta + \lambda(t).\mathrm{Id}\right)^{-1}$$

where Id is the Identity matrix.

The interpretation is the following: at the beginning of the learning process, a high value of factor λ(t) is chosen in order to lead the matrix μ(C) to be diagonal dominant. The rule is therefore close to a first order gradient descent rule.

The factor λ(t) is then decreased in order to be neglected in relation to the approximation of the Hessian part : $\Delta^T\Delta$. At the end of learning, the computation essentially uses the second order information and in a few iterations comes close to the *cost function* minimum.

This presentation of the Levenberg-Marquartd rule shows that backpropragation is necessarily computed in order to estimate the derivatives Δ.

A System Identification

Starting from the previous considerations, the identification of a dynamic system can be addressed by neural networks in computing learning with input-output couples. It is well known that the behaviour of a dynamic system depends not only on external inputs but also on some internal variables that represent the "state" of the system. Under the condition of observability of the system, these state variables are assumed to be past outputs of the real process. However expertise may indicate that another choice may be to select the most relevant state variables (see S. Narendra in [9] for further considerations).

Learning of a discrete-time feedback network

Considering a network at a given instant, learning is performed using the previous external inputs: {i(t)} plus the state variables: the previous output or complementary state variables. Learning on recurrent networks can be performed in at least two ways: the first one consists in taking into account all the previous values using a recurrent method, see for example K. Narendra [9] and P. J. Werbos [10]; the second way takes into account only a few time events, and formulates the backpropagation on a small window of time as proposed by L. Personnaz [11]. The second way was chosen in this study because of its simplicity.

Schemes of identification

Two strategies are possible in order to implement the learning: in the first one the objective is to capture the dynamics of the process. Then the errors coming from the network are taken into account during the learning. The looped input is initialised with the past estimated value of the network. This scheme of identification is called "non directed".

The second way of learning uses measured values coming from the system. This mode is termed "directed".

It is immediately clear that in the case of a neural model with feedback operating on non measurable state variables, the previous discussion is not relevant; the only solution is the non directed model. The identification of the underground flow of water was approached in this way [12].

III NEURAL NETWORKS FOR HYDROLOGY

Because of its complexity there are many models dealing with the rainfall-runoff relation. Usually the models can be classified as: deterministic or statistical; local, global or distributed; static or dynamic; empiric, physics-based or conceptual. Flash flood forecasting is usually addressed by physics-based, conceptual and statistical models. For example, TopModel [13] is a physics-based model used for flood forecasting, while conceptual models have been developed for the same objective: ALHTAÏR [14], MARINE [15] or SCS [16]. For real time use, the models generally have to be distributed [14] [15] [17] or semi-distributed.

Clearly, Neural Networks are statistical models. They have been used for about ten years in an increasing number of applications for elaborate rainfall-runoff models using RBF networks [18], or multilayer networks [19]. Other approaches are also used, such as fuzzy logic [20] or sequential automata [21].

Because of the lack of knowledge about fast floods [22] we hope that neural network models may significantly improve not only flash flood forecasting, but also scientific knowledge about them. This point is at the heart of this work.

A CONTEXT PRESENTATION

The target of our study is the river *Cèze* (fig. 1), a tributary of the *Rhône*. The *Cèze* is 112 km long and its catchment area is about 950 km^2 [23].

The upper part of the river flows on antestephanian schists and gneiss which are impervious; then, on carboniferous deposits (schists, sandstone and coal); in this part of the river, galleries of former coal mines bring water from the neighbouring catchment of the *Avène* river to the *Cèze via the Auzonnet river*. This underground flow is quite low, and had been estimated at 0.5 m^3/s [24]. Then, the *Cèze* flows on oligocen sediments composed of conglomerates which are impervious, before crossing cretaceous deposits, mainly limestones. In this area, the valley flows in a canyon and the plateau has a typically karstic relief. In short, the adjective karstic comes from Slovenia and is generally used for limestone in which the water has eroded galleries. In this part of the river, the relation between the karstic network and the river is not very well understood, so the limit of the area which contributes to outflow in the *Cèze* is not precisely known. The *Cèze* joints the *Rhône* on its right bank at an altitude of 26 meter above sea level, flowing on impervious and semi-pervious deposits.

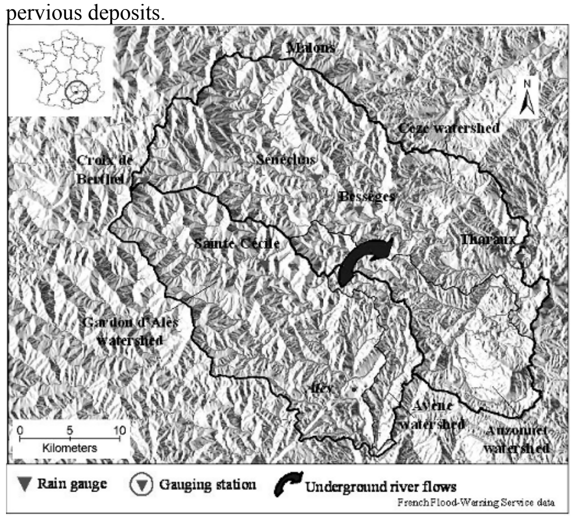

Fig. 1: Location of study area

The main two features of the *Cèze* are its very irregular outflow, which reflects the irregularity of the meteorology, and the contrast between the part of the river on impervious rocks and the part located in a karstic region.

The particular feature of the *Cèz*e is that its flows are not fully explained by the rainfall on its catchment. Several hypotheses have been proposed in order to find another definition of its catchment. One of these hypotheses is that some water could arrive in the river via its affluent the *Auzonnet*, coming via underground circuits from a neighbouring, but different, catchment: the *Gardon d'Alès* river. Two explanations can be found: the karstic network, or galleries of former coal mines. Coal Mines galleries may be neglected because of crumbling in the galleries which limits the flow. We therefore propose in this study to explore the first hypothesis of karstic communication from the *Gardon d'Alès* catchment to the *Cèze* catchment (Fig. 1).

B HYDROLOGICAL DATA

For this study, five floods of the *Cèze* were selected. The following figure shows these events.

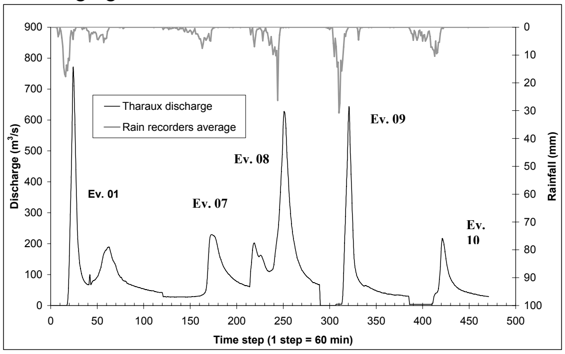

Fig. 2: Floods Hydrographs of the *Cèze* (*Tharaux outlet)*

These data were collected by the flood warning service of the *Gard* Region. The discharge was measured at *Tharaux* gauging station (fig. 1) and the rainfall measured by five rain recorders which located in the *Cèze* and *Gardon d'Alès* catchment areas. For both the discharge and the rainfall, the sampling period was 60 minutes.

TABLE I

Hydrological Database Used

Event	Date	Peak discharge (m^3/s)	Rainfall (mm)
1	22^{th} / 26^{th} October 1993	770	205
2	25^{th} / 28^{th} November 1995	230	105
3	11^{th} / 14^{th} November 1996	630	145
4	06^{th} / 09^{th} October 1997	640	175
5	28^{th} September / 22^{th} October 2000	215	145

Table I contains more details on these events. It can be noticed that because of the difficulty of accurately measuring the rainfall and flows, only five events are available, due to a lack of data for other events.

C FLOOD SIMULATIONS

As usual in the neural network field, the first approach is the multilayered perceptron with one hidden layer. We applied rainfall as inputs and runoff as output. We chose the rainfall measured by 3 rain gauges in the *Cèze* catchment and two in the *Gardon* catchment in order to observe whether the latter input has a major influence on the forecast (Figure 3). An input bias is necessary in order to represent the base flow. Its value is not 1, as is usually applied, but a lower value due to the great number of very low values of the flow during the flood recording. This adjustment is necessary in order not to saturate the sigmoids during learning. The mean of the inputs was shown to be a good value. As shown in Figure 3 we apply the rainfall to the network in a temporal window. This temporal window is essential in order to capture the temporal behaviour of the catchment. Thirty time steps were chosen for rain recorders near the *Tharaux* outlet, and forty five for remore recorders, situated far upstream, in order to take into account a longer propagation time.

At the output of the network we measured the quality of the response using a criterion used in hydrology and called the Nash criterion [25]. The Nash criterion is analogous to the coefficient of determination and is calculated as:

$$\text{Nash} = 1 - \frac{\sum_k \left(o^k - d^k\right)^2}{\sigma^2}$$

where σ is the standard deviation of the test signal.

The Nash criterion takes into accounts the quadratic error and normalises this error by the variance of the signal. The closer the criterion to the value 1, the better the model. If forecasting is limited to predicting the mean value, the criterion is equal to zero; negative values are very bad. Table II contains the values obtained from learning with validation on an example which was not taken into account during learning.

TABLE II

SYNTHETIC RESULT OF THE BEST NASH CRITERIA

Network architecture	Nash criteria computed after test on:					
	Event 1		Event 7		Event 8	
	BP	LM	BP	LM	BP	LM
A	0.68	0.68	0.86	0.92	0.68	0.67
A with Loop	0.75	0.89	0,64	0,88	0,42	0,56
B	0.70	0.74	0.86	0,94	0.62	0.61
C	0.75	0.8	0.89	0.92	0.63	0.66
D	0,75	0.77	0.87	0.92	0.63	0,62

BP is Backpropagation, LM is Levenberg-Marquardt.

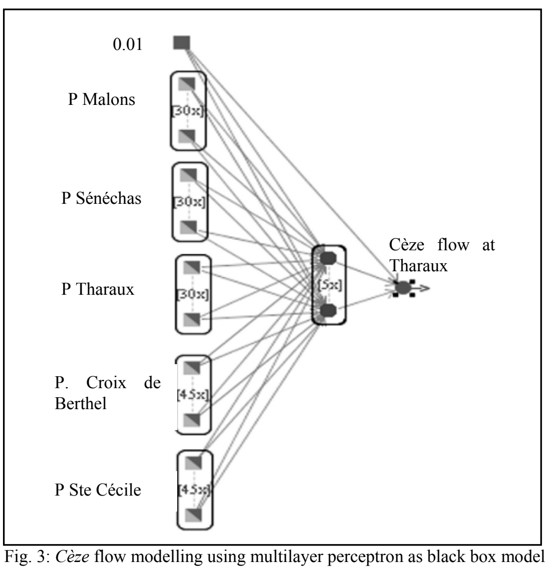

Fig. 3: *Cèze* flow modelling using multilayer perceptron as black box model (Network A.). Each input has a temporal window of 30 or 45 delays

Because of the bad results obtained by this simple static network, we applied the same function, with the same external input, but with a recurrent network. As suggested above, we chose non directed scheme in order better to capture the dynamics of the system. In this case, the Levenberg-Marquardt rule works well and significantly improves the Nash criterion (Table II). It can be noticed that the important estimation of the peak value is not improved by the Levenberg Marquardt. The advantage of the recurrent non directed network is that the input window may be smaller than in the static case. The window of temporal values applied to the networks is ten hours for all the rain recorders. And it is well known that in this type of configuration, the lower the number of coefficients, the better the learning.

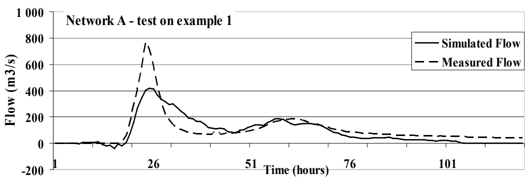

Fig. 4: Measured and forecast flows with network A – BP rule.

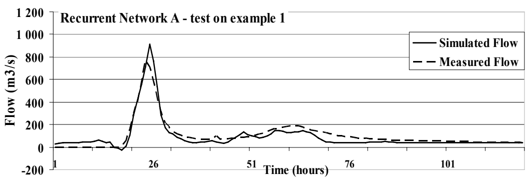

Fig. 5: Measured and forecasted flows with looped network A - LM rule.

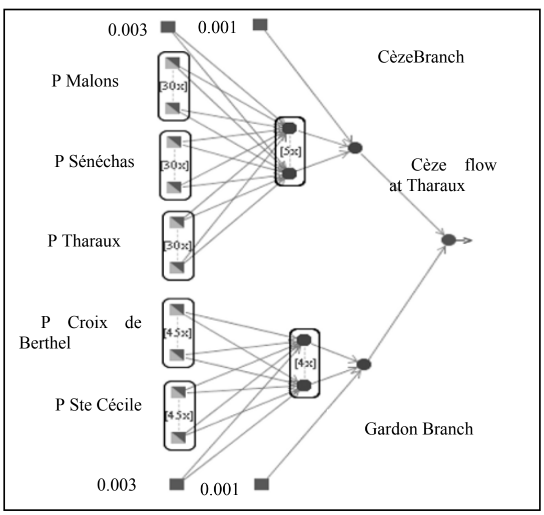

Fig. 6: the contribution of each catchments is computed in a separate branch of the network, and the addition is performed by the output neuron (Network B).

These first simulations allow us to predict the flows accurately but not to estimate the contribution of the *Gardon* catchment. The problem of quantifying the impact of a particularly input on the quality of the prediction is well identified for non-linear models but has no satisfactorily solution. Thus, in order to estimate the contribution of the *Gardon*'s catchment we propose an original architecture which separates the two catchments into two branches: one for the *Gardon* and one for *Cèze* catchment, as shown in Figure 6.

In the proposed architecture, the flow coming from the *Cèze* catchment is computed by a classical network devoted for identification: one hidden layer and one linear output network. Another network devoted to the *Gardon* catchment is computed in parallel. At the end, both networks, called the *Cèze* branch and the *Gardon* branch, are integrated in a single network using a supplementary linear neuron which is the output neuron of the whole network. The flow at *Tharaux* is then computed.

The interest of this network is that the flow coming from each catchment can be obtained by observing the value of the neurons of the second hidden layer: we only need to multiply the output of the neurons by the coefficient linking this neuron to the output neuron. Figure 7 shows the flows obtained in this way with architecture B. In order to illustrate the richness of this approach, we have plotted the estimation of flows on the learning examples, because it is interesting to observe how each sub-basin contributes to the whole flow, for each event.

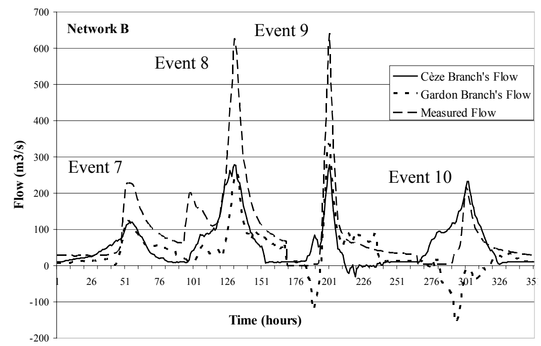

Figure 7: Learning curves obtained for architecture B.

One can note that the *Cèze* branch and the *Gardon* branch contribute equally to the final flow.

The result obtained is that, with architecture B, the contributions of the sub-basins are similar except for event 10. From the hydrological point of view, this result is not possible: the flows coming from the *Gardon* catchment cannot be equal to those from the *Cèze*. Moreover they cannot be negative, as shown for event 10. Thus the architecture is not realistic: the *Croix de Berthel* rain collector should not be inputted to the *Gardon* branch.

Starting from these considerations, we inputted the *Croix de Berthel* rain collector to the *Cèze* branch and left only the *Ste Cécile* rain collector inputting the *Gardon d'Alès* branch, thus building network C.

After learning, we obtained the following results: the forecasting has a Nash criterion of the same order as for network B, but the water now essentially comes from the *Cèze* catchment and only marginal and limited flows come from the

Gardon (Figure 8). We also obtained an estimation of $6.2m^3/s$ for the maximum flow and $2.4m^3/s$ for the mean value.

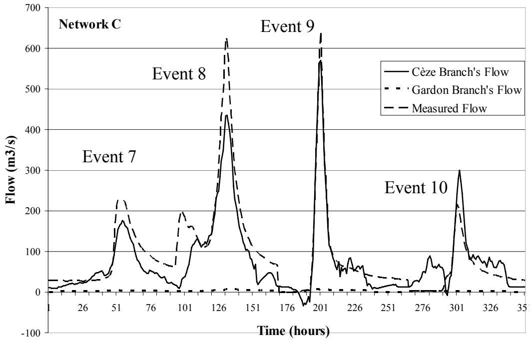

Figure 8: Learning curves obtained for architecture C. On can note that the *Cèze* branch contribute predominately to the final flow. The *Gardon* contribution is very small.

The interpretation is that the flow of the *Cèze* river is well explained by including the *Croix de Berthel* rain collector in the *Cèze* catchment without taking into account a huge karstic flow from the *Gardon*. This interpretation is confirmed by the bibliography: karstic inputs are diffuse along the stream with only one perennial spring: the *Peyrouse* spring. The very interesting property of the neural network is that it estimates this hidden flow.

In fact, the *Croix de Berthel* is near to the frontier of the catchments. Thus, complementarily, we tried an extra network where the *Croix de Berthel* rain collector was connected to both catchments (network D) in order to take into account the possibility of water going to *Tharaux* via the *Céze* or *Gardon* branch. It appears in this case that the results are the same as above (Figure 8): no major quantity of water comes from the Gardon network.

IV CONCLUSION

We have shown in this paper that neural networks can usefully be applied to very complex problems in hydrogeology. We first showed that because of their ability to identify non linear dynamical models, recurrent non-directed neural networks are good candidates for simulating fast floods. Moreover, using a specific architecture we showed that static models can be interpreted in terms of hydrogeology and provide an estimation of hidden variables. This last property is really innovative and opens up a wide field of fruitful research in earth science.

ACKNOWLEDGEMENTS

We are very grateful to Miss S. Eurisouké for the simulations performed during her Master professional training course.

We also wish to thank M. D. Bertin (from SAPI) for his highly affective collaboration in the conception and realization of the Neural Network simulation tool: RnfPro.

We also thank the *Gard* Warning Service for the data used.

REFERENCES

[1]. G. Cybenko"Approximation by Superposition of a Sigmoidal Function". Math. Ctrl Signal Syst, 2 (1989) 293-342.

[2]. A. G. Barto et P. Anandan,."Pattern Recognition Stochastic Learning Automata". IEEE Trans. System Man and Cybernetics. SMC-15 : 360-375 (1985).

[3].Johannet A. and Sarda I. "Goal-Directed Behaviours by Reinforcement Learning". Neurocomputing, Special Issue on Neurap'98. May 1999.

[4].D. Rumelhart, G. Hinton, R. Williams. "Learning Internal Representation by Error Propagation. In "PDP, MIT Press, 1988.

[5] E.A. Bender. "Mathematical Method for Artificial Intelligence".IEEE Computer Society Press. 1996.

[6] A.J. Shepherd. «Second-Order Methods for Neural Networks». Springer 1997.

[7] D.W. Marquardt. Journal of the Society for Industrial and Applied Mathematics, vol 11, pp 431-441.

[8] W.H. Press, S.A.Teukolsky, W.T. Vetterling, B.P. Flannery. «Numerical recipies in C». Cambridge University Press 1992.

[9] Narendra K. S., Parthasarathy K. "Gradient Methods for the Optimization of Dynamical Systems Containning Neural Networks". IEEE trans. neur.net., vol 2, n°2, pp. 252-262, 1991.

[10]Werbos P.J. "Backpropagation Throught Time : What it Does and How to Do It". PROC. IEEE, 78, N°10, PP1550-1560, 1990.

[11].Nerrand O., Roussel P., Personnaz L., Dreyfus G. & Marcos S. (1992) "Neural Networks and Non-Linear Adaptive Filtering : Unifying Concepts and new Algotithms". Neural computation **5**, 165-199.

[12]. A. Johannet, A. Mangin, D. D'Hulst. "Subterranean Water Infiltration Modelling by Neural Networks: Use of Water Source Flow". In Proc. of ICANN 1994; pp1033-1036. M. Marinaro and P.G. Morasso eds, Springer.

[13] K. J. Beven and M. J. Kirby, "A physically based, variable contributing area model of basin hydrology." *Hyd. Sciences*, vol. 24, 1979, pp. 43-69.

[14] P-A. Ayral, S. Sauvagnargues-Lesage, F. Bressand and S. Gay, "Forecasting flash floods with an operational model: application in the South-East of France (Gard)", Flood risk management in Europe: Innovation in Policy and practice". ANTHR, vol. 25. Begum S., Stive Marcel JF, James W (Eds), Springer 2006, 600 p.

[15] V. Estupina-Borrell, J. Chorda, D. Dartus, « Flash floods anticipation." *C. R. Geosciences*, vol. 337, 2005, pp.1109-1119.

[16] E. Gaume, M. Livet, M. Desbordes and J-P. Villeneuve, "Hydrological analysis of the river Aude, France, flash flood on 12 and 13 November 1999". *J. Hydrol.*, vol. 286, 2004, pp. 135-154.

[17] C. Bouvier, P-A. Ayral, P. Brunet, A. Crespy, A. Marchandise and C. Martin, "Recent advances in modelling extreme floods in Mediterranean French catchments",(FRIEND-Amhy) – Extreme events – UNESCO Project, Cosenza (Italy), 3-4 May 2006, 11p.

[18] "H. Moradkhani, K. Hsu, H. V. Gupta, S. Sorooshian. "Improved Streamflow Forecasting Using Self-Organizing Radial Basis Function Artificial Neural Networks. Journal of Hydrology 295 (2004) 246-262.

[19] I. N. Daliakopoulos, P. Coulibaly, I. K. Tsanis. "Groundwater Level Forecasting Using Artificial Neural Networks". Journal of Hydrology 309 (2005) 229-240.

[20] A. P. Jaquin, A. Y. Shamseldin. "Development of rainfall-runoff models using Takagi-Sugeno fuzzy inference systems". Journal of Hydrology 329 (2006) 145-173.

[21] A.-L. Courbis; E. Touraud and B. Vayssade. "Water balance diagnosis based on a simulation tool". ENVIROSOFT'98, 10-12th November 1998, Las Vegas USA,pp199-208.

[22] E. Gaume, P-A. Ayral, C. Bouvier, J.D. Creutin, G. Delrieu and M. Livet, "Hydrological analysis of the Gard River (France) extraordinary flood: 8 and 9 September 2002", 5th Plinius Conference, "Mediterranean storm", 1 – 3 October 2003, 7 p.

[23]E. Dumas: « Statistique géologique, minéralogique, Métallurgique et paléontologique du département du Gard », Paris, ARTHUS BERTRAND, 1875

[24] A. Blachère : « Rapport pour Charbonnages de France », CESAME, 2003.

[25]J.E. Nash , J. V. Sutcliffe. «River Flow Forecasting through Conceptual Models. Part I – A Discussion of Principles » Journal of Hydrologie 10 (1970) 282-290.