Transactions on Information and Communications Technologies vol 19, © 1997 WIT Press, www.witpress.com, ISSN 1743-3517

Application of a Neural Network to transformation inflow-outflow phenomena

O. Giustolisi, M. Mastrorilli, F. Porcaro
Dipartimento di Ingegneria delle Acque, II Facoltà di Ingegneria di Taranto Politecnico di Bari

Abstract

A connectionist model (using a *Neural Network*) applied to the transformation inflow-outflow phenomena of an experimental urban basin has been developed.

It has been made a software, using a Matlab Toolbox, simulating a *Standard Back Propagation Neural Network*, which has been applied to the experimental data of Luzzi's basin. In a previous paper, this application gave encouraging results even if the authors had worked only with twenty five experimentally measured events of the inflow-outflow phenomena.

In the present paper the authors present the results of the application of the Neural Network to the same inflow-outflow phenomena of Luzzi's basin, obtained working with fifty nine experimental inflow-outflow events.

The authors improved the Neural Network efficiency using a variable learning rate and tried to work with inflow-outflow events in the frequency domain in order to obtain better results and faster computation times. To do so, the authors have presented to the Neural Network, inflow-outflow events as complex numbers.

The results can be regarded good considering the complexity of the engineering problem, which is non linear and time dependent.

1 Introduction

The transformation phenomenon of inflow-outflow in a generic basin - urban or natural - is one of the most interesting issues of the Technical Hydrology that summarizes all the hydrological and hydraulic processes which influence the meteoric inflow in each point of the basin producing variable flowrates in the closing section Maione et al.[1].

The present study is a further step of a research Giustolisi et al.[2] which aims at the definition of a connectionist model capable of simulating the transformation phenomenon of inflow-outflow for a quite complex physical system. This aim has been reached by means of pairs pluviogram-hydrogram reordered in the experimental urban basin at Luzzi (Cosenza, Italy) in a three-year period Calomino et al.[3].

The connectionist model simulating the hydrological basin has been realized by means of a Backpropagation Neural Network, Matlab [4] Cammarata [5] [6], characterized by the property to *learn* the solution of a problem by the direct interaction with known events through experimental available pairs of events pluviogram-hydrogram.

Using the standard procedure the physical phenomenon has been acquired by the Neural Network employing about 80% of the inflow-outflow pairs, during the so-called *training phase* or *model set-up*.

The set-up model has been tested on the basis of the remaining 20% of the inflow-outflow pairs to assess the level of reliability of the mathematical model through the evaluation of its *prediction* or *generalization* grade, that is, its capacity of predicting the functioning of the basin system.

Another innovative contribution to the present research, compared with the previous one, is the use of an advanced software: Matlab and its *Neural Network* library installed on a IBM-compatible PC with 166MHz Pentium processor. This allowed a high velocity during the built-in routines, an easier management of the Backpropagation Neural Network, and the graphic visualization of several information during the training phase.

2 Architecture of the Back Propagation Neural Network

In this phase of the research Giustolisi [2] a three layer Neural Network Cammarata [5][6] completely connected to an Error Back Propagation training algorithm and transfer functions associated with non-linear neurons (figure n.1).

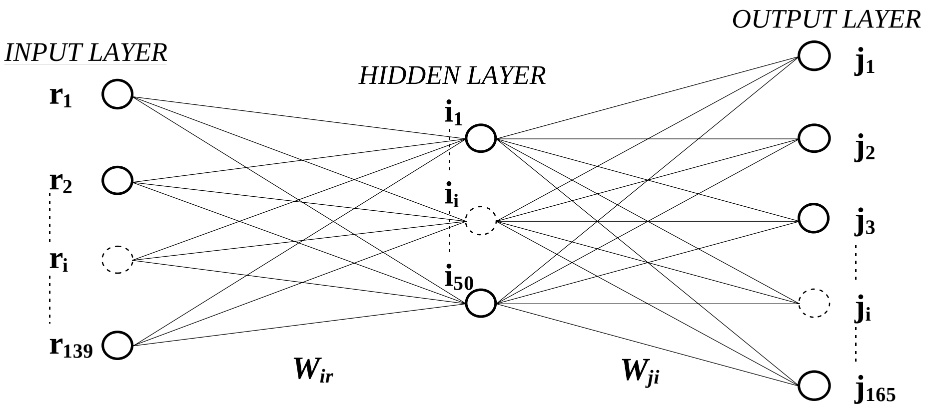

Figure 1

The Neural Network chosen (even though it cannot be the ideal one to simulate the physical phenomenon) allows, after a suitable training, the simulation of very complex physical systems; this feature has been recently regarded sufficient.

The three layers of the Neural Network have been dimensioned as follows:

- input layer: 139 neurons
- hidden layer: 50 neurons
- output layer: 165 neurons

to associate each neuron of the input layer with a value of the experimental pluviogram sampled with an interval of 1 minute, and each neuron of the output layer with a value of the experimental pluviogram also sampled with an interval of 1 minute.

Actually, rainfall have not the same duration and for this reason a number of neuron equal to the number of values sampled of the longest pluviogram (139 minutes) has been employed for the input layer.

Similarly, the number of neurons of the output layer has been fixed equal to the number of the values sampled of the maximum hydrogram (165 minutes). As for the hidden layer, being that there are not methods to determine the suitable number of neurons, a number of 50, experimentally obtained, was accepted. In fact, a lower number (5 and 10 hidden neurons) made the Neural Network unsuitable already during the training phase, whereas a higher number (100 and 200 hidden neurons) did not enable it to generalize in a better way, on the contrary delayed the learning capacity due to the increased calculations.

The choice made is also based upon the fact that the capacity of the Neural Network to simulate the physical phenomenon do not vary substantially with the number of hidden neurons taking into account the uncertainties of the knowledge of the physical system.

The 59 events available were organized in 4 matrixes (table I), whose columns represent each event equaled to the number of components of the correspondent event of maximum duration, adding a suitable number of zeros at the end.

Table I

Name	**Size**	**Notes**
INPUT	139x50	input matrix used for training/set-up
INPUTG	139x9	input matrix used for the generalization/prediction
OUTPUT	165x50	output matrix used for training/set-up
OUTPUTG	165x9	output matrix used for the generalization/prediction

For the set-up the 50 events were not randomly chosen, the first 50 in a chronological order and for the prediction the remaining 9 events were taken. This was aimed at approaching the operating conditions of a trained Neural Network which is to supply hydrograms predicting new rain events whose hydrograms are unknown.

One of the constraint of the Error Back Propagation (EBP) with a constant learning rate is the slowness of the learning process because it requires learning rates not very high to avoid instability problems. Also the choice of the same value of the learning rate is quite difficult for it has never been coded. The software used allowed to solve this problem using the *momentum* and the *variable learning rate* Cammarata [5] Matlab Toolbox [7]:

- The former decreases the Back Propagation sensitivity to little details of the error surface minimizing the risks to be staked in not very deep minima. This happens modifying the law of updating weights as in figure n.1:

$$w_{ij}(t+1) = w_{ij}(t) - \Delta w_{ij} + \beta \cdot \left(w_{ij}(t) - w_{ij}(t+1)\right)$$

where β is a positive constant < 1 termed *momentum* whose function is that of producing a sort of memory of the previous up-dating, standardizing the weight variations; (t) and ($t+1$) are the instant related to a generic learning cycle and the instant related to the cycle immediately after, respectively.

- The latter allows the learning rate to increase and to decrease on the basis of the trend of the global error, thus reducing the training period of the Neural Network since the ideal one is chosen automatically. The training method with a variable learning rate can be written in the following way:

$$SSE'' > SSE' \cdot E_{\max} \Rightarrow \begin{cases} lr'' = lr' \cdot lr_{dec} \\ \beta'' = 0 \end{cases} \quad ; \quad SSE'' < SSE' \cdot E_{\max} \Rightarrow \begin{cases} lr'' = lr' \cdot lr_{inc} \\ \beta'' = \beta' = 0 \end{cases}$$

where:

lr = learning rate

lr_{inc} = increase of learning rate

lr_{dec} = decrease of learning rate

b = momentum constant

E_{max} = maximum increase of error

SSE = quadratic deviation $= \sum_{k=1}^{50} \sum_{j=1}^{165} \left(q_{jk} - \bar{q}_{jk}\right)^2$

The increase and decrease coefficients of the training rate are the most interesting aspects of the software used, that is, the capacity of speeding up the learning when the global error of the *training set* shows a descendent *trend* and slowing down otherwise.

In table II the values of the used variables are listed, which allowed to link the need for a quick learning with the learning efficiency during the whole experimentation.

Table II

lr	**lr_{inc}**	**lr_{dec}**	**b**	**E_{max}**
0,00000001	1,02	0,7	0,1	1,02

For the learning rate we started, for prudential reasons, from a very low value of 10^{-8}, thanks to the capacity of the Neural Network to increase rapidly the learning rate when the global error of the *training set* decreases.

It is well known that the inflow-outflow phenomenon in a hydrological basin is sometimes not linear. It is also known that the choice of an hidden layer of the Neural Network having function of non-linear transferring, makes the choice extremely versatile. On the basis of these considerations, log-sigmoid functions were used for the hidden layer.

The choice of a transferring function for the output was made analyzing two different solutions:

- a log-sigmoid function;
- a linear function.

2.1 Output layer with log-sigmoid transfer function

The choice of a log-sigmoid transfer function also for the output layer supplies values between 0 and 1, that is, always positive; this allowed to avoid the physical absurd of negative output flowrates.

The adoption of a log-sigmoid transfer function has the problem of the *neurons saturation*, when the flowrate values reach the limits of the output domain where the tangent of the function tends to be horizontal and so the correction capacity if the Neural Network is equal to zero Cammarata [5][6].

In addition, the log-sigmoid transfer function has the superior limit, so that a Neural Network also has in output the superior limit.

In our case, the previous limitations give some problems with the characteristics of the physical system under investigation. Putting zero values at the end of each event rain-flowrate the Neural Network presents the output neurons correspondent to the zeros saturated, whereas the need to normalize the experimental hydrograms to be fit in the interval [0,1] of the transfer function of the output neurons creates the need to define a priori a maximum value of flowrate that can be predicted by the Neural Network.

The fact that during the learning phase a Neural Network with log-sigmoid can be more stable due to the particular shape of the function, has as its drawback the need to use, as in our case for the hydrograms, a normalization

$$y_{rete} = \frac{\left(offset + y_{sperimentale}\right)}{normal}$$

where the *offset* different from zero would serve to eliminate the saturation problem caused by the zero values of the hydrograms ; and the large *normal* would eliminate the problem of the maximum value predicted.

Lastly, with the choice made, if properly carried out, the output neurons would work in the area where the log-sigmoid function is equal to 0,5 and so in the linear segment (in good approximation).

2.2 Output layer with linear transfer function

The choice of linear transfer functions allows, on the one hand, to avoid neurons saturation problems, on the other hand, generates values of flowrate also negative. However, once the learning phase was finished, this problem resulted of neglectable proportions, since the Neural Network learns to supply flowrate values almost always positive.

The adoption of a linear transfer function for the output neuron layer solves saturation problems and also problems related to the fact that the transfer function has the superior limit. A normalization of the outputs in the interval [0,0.4] was carried out though not necessary, using the equation previously introduced, with *offset*=0 and *normal*=y_{max} */0,4.* The same was done for the input data of rain in the normalization interval [0,0.4] to compare the results obtained with those of other basins and to minimize saturation phenomena of the intermediate layer when a log-sigmoid is chosen[1].

The minor stability of the linear transfer function has been solved initializing the weights and the biases with low values, as we shall see, and adopting a very small momentum β.

3 Description of the Neural Network

Indicated with <*r*> the generic neuron of the *input layer*, with <*i*> the generic neuron of the *hidden layer*, consisting in *p* neurons (*p*=50 in the present case), and with <*j*> the generic neuron of the *output layer*, the Neural Network transforms an input vector e_r (139 elements) into an output vector y_j (165 elements) through synaptic weights (w_{ir};w_{ji}) and biases (θ_i;θ_j), connecting the input-hidden and the hidden-output layers, and the hidden layer represented by vectors x_i according to eqs. Giustolisi et al. [2] Cammarata [5][6]:

$$P_i = \sum_{r=1}^{139} w_{ir} \cdot e_r - \theta_i \qquad P_j = \sum_{i=1}^{50} w_{ji} \cdot x_i - \theta_j$$

$$x_i = g_1(P_i) \qquad y_j = g_2(P_j)$$

selected as transfer functions:

$$g_1(P) = \frac{1}{1+e^{-P}} \qquad \text{(log-sigmoid)}$$

$$g_2(P) = P \qquad \text{(linear)}$$

The flowrate al jth minute of kth event calculated by the Neural Network is given by:

$$\nu_j = F_j(X_K, W) = g_2\left[\sum_{i=1}^{50} W_{ji} \cdot g_1\left(\sum_{r=1}^{139} W_{ir} \cdot e_r\right)\right]$$

---

[1] The constant normalization allow us to compare the starting learning rate, the starting wieghts and SSE between trials.

Lastly, the initialization of the Neural Network has been carried out simply using the *rands*[2] function of MATLAB:

$$a = 0{,}01 \cdot [rands]$$

which allowed to randomly assign to weight and biases values in the range of [-*0,01*;+*0,01*] to avoid saturation problems of the transfer functions examined in the previous paragraph.

4 Input-output pairs in time domain

The 50 pairs of input-output events used by the Neural Network in set-up phase (training phase) were presented many times in several ways to obtain the optimization of the Neural Network answers during the prediction phase (generalization) of 9 pairs of input-output events.

In fact, the pairs available for the Neural Network training are related (table III) to events of inflow-outflow featured by quite variable water volumes, and also the maximum and mean values of rainfall and flowrates, lasting from few minutes up to 165 minutes.

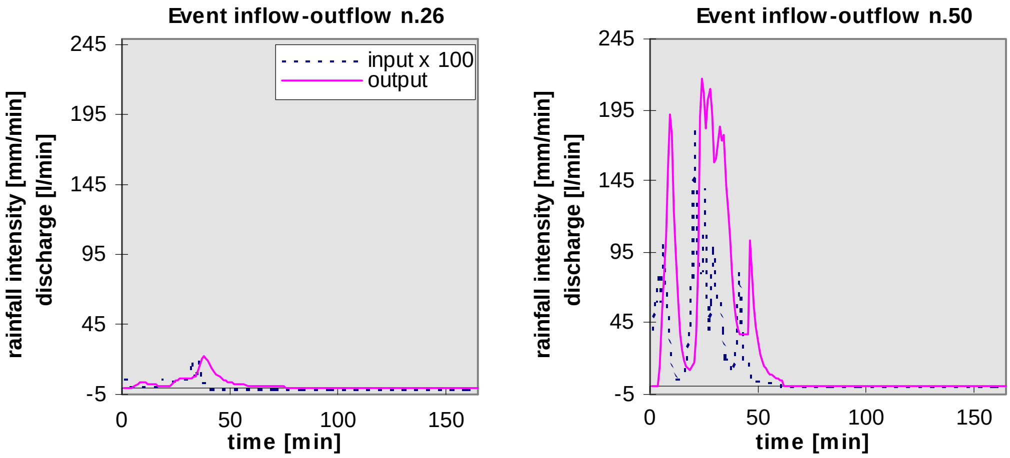

Figure 2

The rainfall flowing into the basin is not equal to the downflow flowing out from the same basin since only gross rainfalls are experimentally available, as recorded by the pluviographs, which are not reduced of the volume that thanks to the basin configuration does not contribute to the formation of a downflow in the monitored section of the Neural Network of urban drainage, but is lost due

[2] The function ***rands(S,R)*** of Neural Network Toolbox gives us the SxR matrix of the weights W and it gives us the Sx1 matrix of the biases θ with random elements in the range [0,1].

Transactions on Information and Communications Technologies vol 19, © 1997 WIT Press, www.witpress.com, ISSN 1743-3517

to infiltration into subsurface strata, to evapo-transpiration, into little storage capacities, or for other physical causes Maione et al.[1].

Table III

N. event	Date	t min	total h mm	mean i mm/h	volume m^3	flux coeff.
1	27/11/87	18	3.6	12.0	67.9	0.41
2	27/11/87	43	3.8	5.3	71.7	0.44
3	09/12/87	139	7.8	3.4	147.1	0.46
4	14/12/87	36	7.2	12.0	135.8	0.31
5	23/01/88	72	7.8	6.5	147.1	0.49
6	23/01/88	110	5.6	3.1	105.6	0.70
7	14/09/88	38	8.6	13.6	162.2	0.52
8	15/09/88	40	13.8	20.7	260.3	0.73
9	21/09/88	53	2.4	2.7	45.3	0.63
10	08/10/88	13	3.0	13.8	56.6	0.45
11	08/10/88	59	4.6	4.7	86.8	0.48
12	20/11/88	94	4.8	3.1	90.5	0.62
13	20/11/88	53	2.6	2.9	49.0	0.60
14	21/11/88	25	1.8	4.3	33.9	0.53
15	21/11/88	15	2.0	8.0	37.7	0.49
16	21/11/88	69	5.2	4.5	98.1	0.59
17	21/11/88	107	8.8	4.9	166.0	0.67
18	22/11/88	42	3.8	5.4	71.7	0.51
19	22/11/88	111	5.6	3.0	105.6	0.59
20	23/11/88	12	1.0	5.0	18.9	0.45
21	23/11/88	67	3.8	3.4	71.7	0.81
22	24/11/88	17	1.4	4.9	26.4	0.68
23	25/02/89	48	6.6	8.3	124.5	0.59
24	26/02/89	12	1.4	7.0	26.4	0.59
25	26/02/89	44	4.0	5.5	75.4	0.55
26	27/02/89	40	2.0	3.0	37.7	0.51
27	27/02/89	62	4.2	4.1	79.2	0.59
28	27/02/89	60	3.4	3.4	64.1	0.79
29	27/02/89	32	1.6	3.0	30.2	0.78
30	28/02/89	67	4.6	4.1	86.8	0.81
31	28/02/89	127	6.0	2.8	113.2	0.76
32	28/02/89	18	1.0	3.3	18.9	0.83
33	28/02/89	40	1.2	1.8	22.6	0.66
34	02/03/89	59	4.8	4.9	90.5	0.68
35	02/03/89	60	5.4	5.4	101.8	0.72
36	02/03/89	12	1.8	9.0	33.9	0.78
37	04/03/89	4	1.0	15.0	18.9	0.39
38	04/03/89	44	7.4	10.1	139.6	0.67
39	04/03/89	29	2.6	5.4	49.0	0.75
40	13/04/89	14	1.6	6.9	30.2	0.55
41	13/04/89	26	1.2	2.8	22.6	0.50
42	14/04/89	39	3.2	4.9	60.4	0.63
43	14/04/89	44	5.2	7.1	98.1	0.72
44	14/04/89	65	7.4	6.8	139.6	0.78
45	15/04/89	7	1.8	15.4	33.9	0.74
46	01/05/89	39	3.2	4.9	60.4	0.75
47	02/05/89	73	4.0	3.3	75.4	0.77
48	02/05/89	137	7.8	3.4	147.1	0.79
49	18/05/89	14	2.2	9.4	41.5	0.33
50	18/05/89	59	22.6	23.0	426.2	0.66
51	12/06/89	31	3.6	7.0	67.9	0.41
52	17/06/89	22	6.6	18.0	124.5	0.56
53	26/03/90	37	8.8	14.3	166.0	0.54
54	29/03/90	25	2.2	5.3	41.5	0.74
55	30/03/90	37	3.2	5.2	60.4	0.80
56	04/04/90	89	4.8	3.2	90.5	0.71
57	09/04/90	23	2.2	5.7	41.5	0.53
58	10/04/90	393	26.6	4.1	501.7	0.76
59	10/04/90	80	2.4	1.8	45.3	0.77

To avoid the problem of the different durations of the inflow-outflow events it has been necessary to uniform the length adopting a number of *input* neurons equal to the maximum number of minutes of the longest rain (139 minutes) and a number of *output neurons* equal to the maximum number of minutes of the longest downflow (165 minutes). This criteria has been adopted for the normalization of the lengths of input-output sequences to maintain this information within the pairs of events (figure n.2).

Consequently, observing the curves of the 59 input-output of the events used in the various phases of set-up of the connectionist model of simulation of the physical system of the urban basin at Luzzi as sequences discrete in time, the appear of very variable order of magnitude and affected by long series of zero values at the end.

In the Back Propagation the equation which regulate the correction of weights during the learning are:

$$\Delta W_{ji} = -\eta \cdot \frac{\partial E_k}{\partial W_{ji}} = -\eta \cdot \left(y_j - \bar{y}_j\right) \cdot f'\left(P_j\right) x_i = \eta \cdot \left(y_j - \bar{y}_j\right) \cdot e^{-P_j} \cdot x_i \cdot y_j^2$$

$$\Delta W_{ir} = -\eta \cdot \frac{\partial E_k}{\partial x_i} \cdot f'\left(P_i\right) \cdot e_r = -\eta \cdot \left(x_i - \bar{x}_i\right) \cdot e^{-P_i} \cdot e^r \cdot x_i^r \qquad (1)$$

where *r, i, j* are the indexes of the input, hidden and output layers, respectively. Eq. (1) is directly proportional to the input e^{Pj} and the output y_j; therefore, the corrections will be the more remarkable, the higher the inflows (rain input) and the outflows (flowrate output) will be. This is an anomalous learning which gives a different weight to the events to recorder.

The set-up in Giustolis et al.[2] of the Neural Network with the pairs of inflow-outflow in time domain, made of 20+5 pairs available between model set-up and its validation, supplied satisfactory results considering the modest number of experimental measurements compared with the theoretical need of connectionist models.

Now, having a larger number of pairs of experimental events, the training process has been repeated to point out the improvements of the connectionist model deriving from the increased width of the learning set available and to a greater expertise acquired on the functioning of the Neural Networks.

In the present paper, in fact, the Neural Network used, similar to the one used in Giustolisi et al.[2], presents some innovative technical solutions adopted in the learning phase as pointed out in the previous chapter, and as for its architecture, the linear transfer function of the output layer chosen to avoid saturation probelms during the learning caused by the high number of zeros at the end.

5 Input-output pairs in frequency domain

As new solution, an alternative to the presentation in time domain of the inflow-outflow pairs has been illustrated.

To this aim a Neural Network, similar to the previous one, has been tested presenting experimental pairs in the frequency domain, that is, as sum of circular functions.

Such a solution has been adapted to better the efficiency of the simulation model through a specific pre-elaboration of the set-up data. This elaboration of the 50 pairs of experimental events was aimed at supplying information to the Neural Network in a better shape to its comprehension without loosing information in this step.

The idea to present to the connctionist model pairs of rainfall-flowrate events in the frequency domain is originated by the consideration based upon a specific result of the theory of linear systems invariant with translation, which should be pointed out.

It is known that a linear system invariant with translation Oppenheim et al. [8], in this discrete case, can be characterized in time domain by the answer to the unitarian impulse δ(n) in the following way,

$$h(n) = T[\delta(n)]$$
$$\mathrm{y(n)} = \sum_{k=-\infty}^{\infty} x(k) \cdot h(n-k) \tag{2}$$

where h(n) is the answer to the unitarian impulse of the T[] system and y(n) is the output of the system generated by the generic input x(n).

This system is, then, completely described with a sum of convolution, second equation in (2), once the answer to δ(n) is known.

The theory of linear systems invariant to the translation allows to describe in the frequency domain the sequences in time x(n), y(n), h(n) by means of the Discrete Fourier Series DFS Oppenheim et al.[8] obtaining the following equations

$$
\begin{aligned}
H(e^{i\cdot\omega}) &= \sum_{n=0}^{\infty} h(n)\cdot e^{i\cdot\omega\cdot n} \\
Y(e^{i\cdot\omega}) &= \sum_{n=0}^{\infty} y(n)\cdot e^{i\cdot\omega\cdot n} \\
X(e^{i\cdot\omega}) &= \sum_{n=0}^{\infty} x(n)\cdot e^{i\cdot\omega\cdot n} \\
Y(e^{i\cdot\omega}) &= H(e^{i\cdot\omega})\cdot X(e^{i\cdot\omega})
\end{aligned}
\qquad (3)
$$

which are valid provided that the series in (3) are convergent and then x(n), y(n), h(n) can be summed:

$$
\sum_{n=0}^{\infty} |x(n)| < \infty \quad \sum_{n=0}^{\infty} |y(n)| < \infty \quad \sum_{n=0}^{\infty} |h(n)| < \infty
$$

This means that a linear system can be described in frequency domain in terms of products of complex numbers ($e^{i\cdot\omega}$) and $H(e^{i\cdot\omega})$ which represent x(n) and h(n) in terms of sum of cosines functions of pulses ω, to obtain complex numbers $Y(e^{i\cdot\omega})$.

The comparison between the relations of the linear systems in time domain and the linear systems in frequency domain points out that in the second case the description of the system is made easier being based on the product of complex numbers.

This last result is obtained for linear systems invariant with translation, even if the connectionist model is meant to simulate a physical system affected by high non-linearities and does not answer to the property of invariance with translation of the answer to the unitary pulse (Input Unitary Hydrogram), is essential to consider that a Neural Network working into the frequency domain could give better results than a Neural Network working into the time domain.

5.1 Characteristics of the Neural Network in frequency domain

Based on the considerations of the previous chapter the 50 pairs of inflow-outflow events have been pre-elaborated to obtained a description as the sum of circular functions according to DFS before their presentation to the Neural Network for the model set-up.

It has been confirmed the choice of giving to the shorter sequences a number of zeros to normalize the lengths of the events of inflows and outflows calculated by the model at the maximum duration of the experimental data available N_{inp}=139 and N_{out}=165; the events, then, appear as finite and limited sequences, which can be summed in a model, according to which DFS could be calculated as:

$$\left[X(e^{i\cdot\omega})\right]_j = \sum_{n=0}^{Ninp-1}\left[x(n)\right]_j \cdot e^{-i\cdot\omega\cdot n} \qquad \text{for } j=1,2,\ldots..,50 \text{ inflows}$$
$$\left[Y(e^{i\cdot\omega})\right]_j = \sum_{n=0}^{Nout-1}\left[y(n)\right]_j \cdot e^{-i\cdot\omega\cdot n} \qquad \text{for } j=1,2,\ldots..,50 \text{ outflows} \tag{4}$$

The calculation of (4) has been carried out using first the Discrete Fourier Transformed DFT Oppenheim et al.[8],

$$\left[X(k)\right]_j = \frac{1}{N_{inp}}\cdot\sum_{n=0}^{Ninp-1}\left[x(n)\right]_j \cdot e^{-i\cdot(2\pi/Ninp)\cdot n\cdot k} \qquad k=1,\ldots,(\mathrm{N_{inp}}\text{ -1}) \text{ e } j=1,\ldots,50$$
$$\left[Y(k)\right]_j = \frac{1}{N_{out}}\cdot\sum_{n=0}^{Nout-1}\left[y(n)\right]_j \cdot e^{-i\cdot(2\pi/Nout)\cdot n\cdot k} \qquad k=1,\ldots,(\mathrm{N_{out}}\text{ -1}) \text{ e } j=1,\ldots,50 \tag{5}$$

and Fast Fourier Transformer non power of two alghoritm.

$[X(k)]_j$ e $[Y(k)]_j$ calculated with (5) are formed by (N_{inp} -1) and (N_{out} -1) complex conjugate numbers, for each of the 50 events j, related to (N_{inp}-1)/2=67 and (N_{out}-1)/2=82 circular functions to which two complex number are added having the sole real part different from zero, related to the continuous component given by the particular shape of (4) or (5) for ω=0 or k=0:

$$\left[X(0)\right]_j = \frac{1}{N_{inp}}\cdot\sum_{n=0}^{Ninp-1}\left[x(n)\right]_j \qquad \text{per } j=1,2,\ldots..,50$$
$$\left[Y(0)\right]_j = \frac{1}{N_{out}}\cdot\sum_{n=0}^{Nout-1}\left[y(n)\right]_j \qquad \text{per } j=1,2,\ldots..,50 \tag{6}$$

Eqs. (6) point out how the information supplied to the Neural Network linked to inflows and outflows is contained in the first value of the complex sequences which represent the mean value of x(n) and y(n).

In fact in eqs. (6), a meno terms of constant proportionality $1/N_{inp}$ and $1/N_{out}$, the sums represents exactly the inflows and the outflows since the specific mean value is obtained not as a function of the effective duration of the events but as function of the values of maximum duration N_{inp} and N_{out}.

From the point of view of its architecture, the Neural Network in the frequency domain does not differ much from that in the time domain; in fact, the latter is formed by N_{inp} and N_{out} input and output neurons, respectively, with the same number of hidden neurons and the same learning rate and momentum of the neural net previously adopted.

As for the transfer functions, two linear functions were used to avoid whichever risk of saturation, being aware of the fact that in the frequency domain a completely linear Neural Network in which $g_1(P)=g_2(P)=P$ can simulate a non linear phenomenon since the input frequency is connected to all the output.

A completely linear Neural Network also allows to minimize the importance of initial W_{ir} and W_{ji}, to increase the learning speed and does not require the normalization.

To the new Neural Network, on the contrary, the continuous component and the real and the imagery parts correlated to $(N_{inp}-1)/2=67$ circular functions of description of each of the 50 $[X(e^{i\cdot\omega})]_j$ were presented in the input neurons N_{inp}, in ordinate sequence.

Similarly, the continuous component and the real and the imagery parts correlated to $(N_{out}-1)/2=82$ circular functions of description of each of the 50 $[Y(e^{i\cdot\omega})]_j$ were presented to the output neurons N_{out}, in ordinate sequence.

Such a Neural Network is capable of grasping the most important consequence of (6) linked to the fact that the information on the volume exiting the physical system, the hydrological basin at Luzzi, for the single event j of model set-up before and after the assessment of the predicting capacity of the model in a second time is contained in one single output value.

The Neural Network in the frequency domain, after a suitable set-up, should be capable of recognize the outflow volume, contained in only one neuron of the model, by the recording of the inflow volume, in only one input neuron of the model, and of the other input neurons linked to the circular components defining the shape of the rainfall pluviogram and also its duration.

This structure of the Neural Network is reasonable from a physical point of view since the water volume lost by the hydrological basin can be considered dependent both on the global water volume and on the pluviogram shape.

The non linearity of the Neural Network answer resides in the fact that imagining an equal number of input and output neurons to have an equal interval of sampling of the inflow and outflow sequences, so that the axis of the frequencies of $[X(e^{i\cdot\omega})]_j$ and $[Y(e^{i\cdot\omega})]_j$ would be of equal resolution, one would not obtain a constant answer $Y(e^{i\cdot\omega})/X(e^{i\cdot\omega})$ of the model being w given.

The remarkable non linearity of the hydrological basin, on the contrary, is easily assumed from the graphic (figure n.3) of the values of the ratios of the inflows to the outflows as in table III, and from the relative coefficient of outflow which are not constant.

The Neural Network set up with the data of the frequency domain allows to solve the conceptual problem deriving from the fact that the total connection among the neurons of a Backpropagation Neural Network implies a functional connection also among the neurons in the input and output layers, where the former can physically correspond to minutes of rain falling after the outflows represented by the latter.

This situation is not physically reasonable, even though it does not influence the good functioning of the Neural Network in the time domain Giustolisi et al.[2] since in the connectionist paradigm the possibility the weights adjust to the situation assuming suitable values, is included.

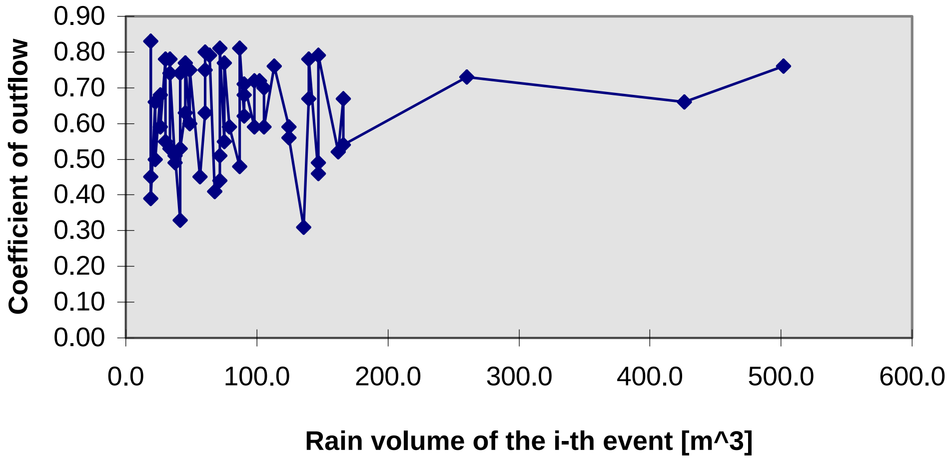

Figure 3

What previously said, can influence the number of neurons of the hidden layer, and the total 0 weights of the Neural Network, since part of the functional connections, as already said, could a priori be eliminated.

To conclude, from a schematic point of view, the Neural Network trained in the frequency domain works as shown in figure n.4. The pre-elaboration of x(n) data by means of DFT was followed by a output decoding by means of IDFT to obtain y(n) that is a sequence still in the time domain. This last application is also useful in the engineering field.

Figura n.4

6 Analysis of results

The learning process was controlled observing the curves volute and calculated both in memorization and in generalization. This allowed to assess critically the capacity of learning of the Neural Network and, when this does not occur, to intervene on the main coefficients or on the architecture of the same Network.

To have a more objective criterion of evaluation, during the learning, a suitable expression of the error has been chosen, which would be synthetic as a number and significant as a graph. Various criteria exist to compare hydrograms recorded and those simulated by the model, which allow to evaluate the model performances with respect to the peak flowrate or to the hydrogram shape, that is, with respect to the outflows or to the duration of the peak flowrate Calomino et al.[9].

It should be remembered that the Neural Network approaches the problems in terms of recognition of the curve shape and does not focus on the values of the peaks.

Coherently with the engineering objectives of the model, to measure the accuracy in the recognition of a generic hydrogram, the percentage error of the inflow related to the single event defined as follows:

$$E = 100 \; (\Sigma|q_{j\text{-}sim} - q_{j\text{-}oss}| \;) \; / \; Q_{medio}$$

has been assumed, where Q_{medio} represents the effective mean value of the flowrate, and so without the burden of the zeros at the end of each event. This choice has been suggested by the need not to influence the value of the error of the rain duration.

As predicted by the software, the learning has been guided by the minimization of the global error SSE and has been carried to a very large number of cycles saving the Network configuration (weights matrix) to which the minimum value of the error E on the validation-set corresponded.

Table IV-a

TIME DOMAIN	*Events*	*Error (%)*										*Mean Error (%)*
Training-set	***1 - 10***	12	16	32	9	11	22	2	2	11	9	
	11 - 20	20	20	12	11	9	10	9	14	26	28	
	21 - 30	16	11	6	13	12	26	11	14	13	16	**13,15**
	31 - 40	26	20	14	16	8	10	18	6	14	13	
	41 - 50	15	6	7	4	10	8	15	10	13	1	
Validation-set	***1 - 9***	33	17	11	12	30	17	15	22	45		**22,44**

Table IV-b

FREQUENCY DOMAIN	*Events*	*Error (%)*										*Mean Error (%)*
Training-set	***1 - 10***	10	14	26	6	10	18	3	2	12	7	
	11 - 20	16	21	14	14	8	9	8	14	24	23	
	21 - 30	15	12	5	14	11	21	11	11	15	16	**12,15**
	31 - 40	18	22	15	13	7	11	14	5	15	11	
	41 - 50	15	6	7	4	9	9	16	7	12	1	
Validation-set	***1 - 9***	32	17	11	14	24	17	14	24	41		**21,53**

Tables IV-a and IV-b show this value of the error for each event both during the learning and the generalization in both domains of the Neural Network.

6.1 Time domain

Operating on a package of inflow-outflow events much larger than the one in Giustolisi et al.[2] the Network could *learn* better the particularities of the physical phenomenon and this lead to an increase of the predicting capacities of the network. This result confirms the fact that having a *training-set* and a *validation-set* of adequate sizes, and so sufficiently large, the performances of the Neural Network on both sets tend to be the same.

The improvements in the generalization are more evident comparing the sole two events of the training-set in common between the present investigation and Giustolisi et al.[2] (figure n.5):

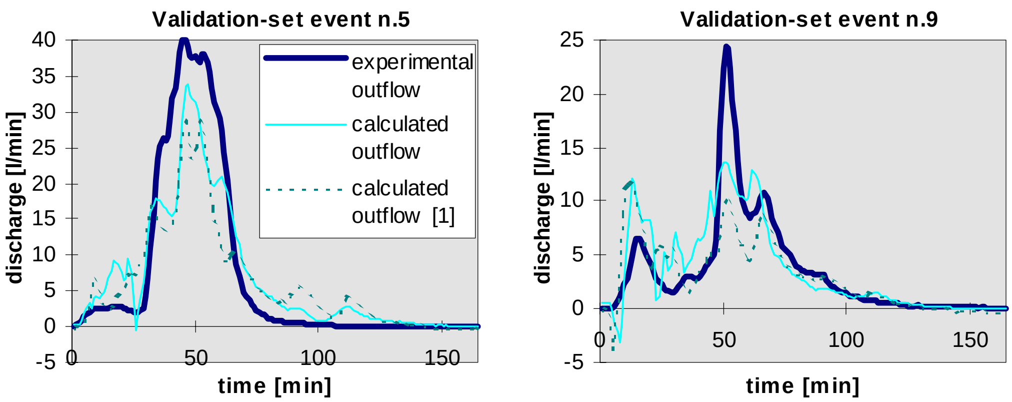

Figure 5

Although the prediction of these two events resulted quite difficult in both cases, a better prediction of the peak flowrate and of the shape of the hydrogram can be seen. The hydrogram presents isolated negative values which, though not coherent with the physical reality, do not deprive to curves of their significance.

6.2 Frequency domain

For each of the 9 events of the *validation-set* and for one of the *training-set*, the one characterized by the maximum error, a graph is presented comprising two curves to compare the hydrogram given and the one calculated by the Neural Network (figure n.6).

The performances of the Neural Network in generalization are to be considered quite interesting since it showed to be capable of identifying the shape of the curves and because of the accurate determination of the instants of the main peaks of the hydrograms.

As for the prediction of the peak flowrate, instead, it can be good for events 2-3-4-5-6-8; the previous hydrograms calculated presented a trend different from the real one in the last segment of the flooding wave of those hydrograms characterized by minor duration.

As for the events 1-7-9, instead, though the prediction of the peak was assessed sufficient, the hydrograms calculated by the Neural Network are not as good as the previous ones.

During the set-up the Neural Network cannot, for the particular type of algorithm of learning used, give equal importance to all events rain-flowrate used for its set-up and focuses on the bigger ones both as outflow volumes and maximum values of the flowrates often with respect to the water volumes. It is difficult, in fact, to generalize the events 1-7-9 which are the three events of the *validation-set* characterized by the lowest values of the maximum flowrate and of the outflow volumes.

This last circumstance is anyhow acceptable from an engineering point of view because the precise simulation of inflow-outflow events characterized by the larger parameters is technically more interesting.

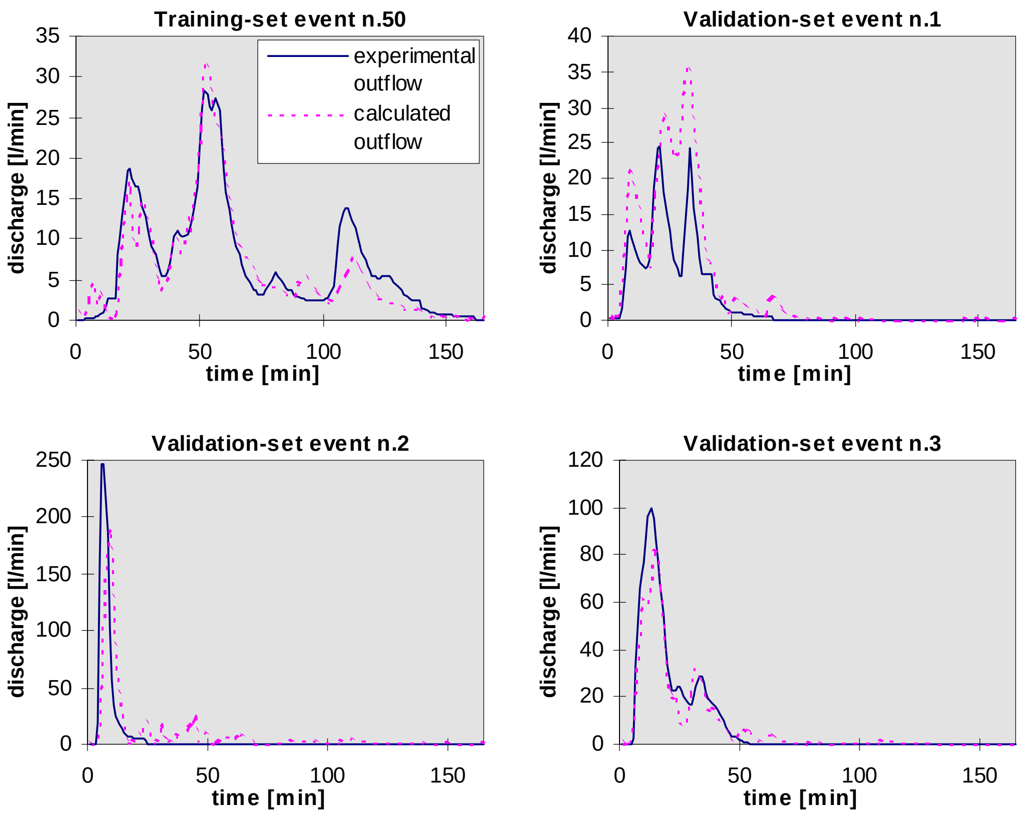

Figure 6

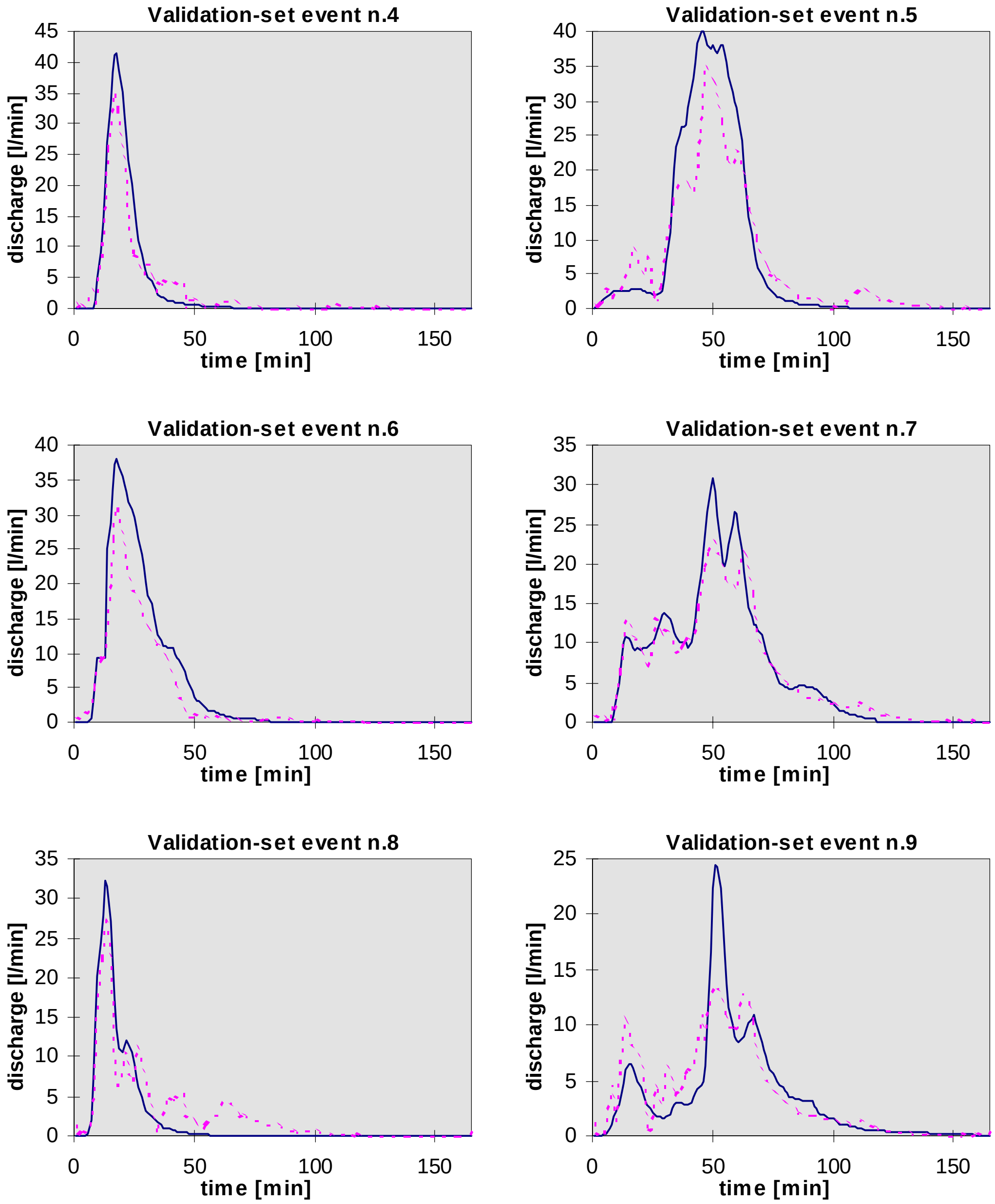

Figure 6

7 Conclusions

The present paper confirmed the results of Giustolisi et al.[2] pointing out an improvement of the predicting capacity of the connectionist model, as anticipated at a theoretical level, thanks to the availability of a larger set of events used for its setting up.

The connectionist model working in the frequency domain produced an increase, even if not remarkable, of the capacity of the Neural Network to simulate the urban basin, that is a real physical situation. This network is much quicker in the set-up phase due to its property to simulate a non linear physical phenomenon also assigning linear transfer functions to its hidden and output neurons. The results obtained are encouraging considering that the connectionist model used is usually aimed at the recognition of shapes, whereas in the physical situation under investigation the input-output events are time series.

The Authors will, then, approach a second research on the achievement of different results:

- simulation of other experimental basins by means of connectionst models to generalize the results obtained in the present investigation also considering that different basins could have different levels of complexity;
- acquisition of structures of the Neural Network capable of supplying better simulation as a function of the particular physical system under investigation;
- comparison between the simulation connectionist models and those of identification of linear systems;
- employment of simulation model as support for the knowledge of the physical characteristics of the urban basin whose data are partly implicitly contained in the events used to set up the mathematical model.

The choice of the Neural Network architecture, in particular of the number of neurons of its hidden layer, is not the critical point during the set-up of the connectionist model by means of the experimental data of the urban basin. This particular situation derives from the circumstance that small differences of the predicting capacity of the model as function of the variation of its architecture have been evaluated also considering the uncertainties proper to the knowledge of a physical system through experimental input-output curves only.

Key Words

1.Neural Network
2.Hydrology
3.Urban Drainage
4.Urban Basin

References

[1] U. Maione, A. Brath: "La sistemazione dei corsi d'acqua naturali", Editoriale Bios.

[2] Giustolisi O., Mastrorilli M.: "Realizzazione di un modello connessionista applicato ad un problema di Idrologia Urbana", XXIV Convegno di Idraulica e Costruzioni Idrauliche - Napoli 2c2-09-1994.

[3] Calomino F., Caputo V., Galasso L. e Piro P. - Dipartimento di Difesa del Suolo Università della Calabria - "Il bacino sperimentale urbano di Luzzi (CS)" - *Osservazioni sperimentali nel periodo 1987-1990*, Editoriale Bios s.a.s.

[4] Matlab software, rel.4.2.1c, The Math Works Inc., 1992.
[5] Cammarata S. "Reti neuronali" - *Una introduzione all' "altra " intelligenza artificiale*, Etaslibri, 1990.
[6] Cammarata S. "Sistemi fuzzy" - *Un'applicazione di successo dell'intelligenza artificiale*, Etaslibri, 1990.
[7] Neural Network Toolbox, for Matlab, The Math Works Inc., 1994.
[8] Oppenheim A.V., Schafer R.W. "Elaborazione numerica dei segnali" - editrice FrancoAngeli - Milano 1990.
[9] F. Calomino, C. Maksimovic e B. Molino: "Urban Drainage" - *Experimental catchments in Italy*, Editoriale Bios.