Nonlin. Processes Geophys., 13, 443–448, 2006
www.nonlin-processes-geophys.net/13/443/2006/

Nonlinear Processes in Geophysics

Flash-flood forecasting by means of neural networks and nearest neighbour approach – a comparative study

A. Piotrowski, J. J. Napiórkowski, and P.M. Rowiński

Institute of Geophysics, Polish Academy of Sciences, Warsaw, Poland

Received: 3 April 2006 – Revised: 8 June 2006 – Accepted: 21 August 2006 – Published: 25 August 2006

Abstract. In this paper, Multi-Layer Perceptron and Radial-Basis Function Neural Networks, along with the Nearest Neighbour approach and linear regression are utilized for flash-flood forecasting in the mountainous Nysa Klodzka river catchment. It turned out that the Radial-Basis Function Neural Network is the best model for 3- and 6-h lead time prediction and the only reliable one for 9-h lead time forecasting for the largest flood used as a test case.

1 Introduction

One of the most dangerous flood events which happened in Poland took place in July 1997, severely affecting the southern and western part of the country. The small but densely populated catchment of the Nysa Kłodzka river, a tributary of the Oder river, suffered tremendously. The area of the Nysa Kłodzka catchment (described in more detail in Napiórkowski, 2003) is about 4565 square kilometres, with mountainous relief, its highest peak reaching 1440 m above sea level. During events of high precipitation, steep valley slopes may shorten the catchment response time to hours, making proper forecasts much more difficult and giving local people less time for preparation.

Rainfall-runoff modelling may be performed by means of data-driven methods, which do not describe physical processes (Napiórkowski and Piotrowski, 2005). Among the most useful techniques are Multi-Layer Perceptron Neural Networks (MLP), Radial-Basis Function Networks (RBF) and the Nearest Neighbours approach (NN). In the present study, these approaches, along with linear regression, are applied for flash-flood forecasting, based on a small data set of rainfall and runoff measurements collected during six flood events that have afflicted the Nysa Kłodzka catchment since 1965. Although the mentioned techniques are well known in the hydrological sciences, it is rare that all of them are applied to the same task, which may help in a comparison of their applicability to practical problems. Literature examples comprise of the works of Brath et al. (2002) using MLP networks and the NN approach for precipitation forecasting, Jayawardena and Fernando (1998), Dawson et al. (2002), Moradkhani et al. (2004) and Wang et al. (2006) applying MLP and RBF networks for flow prediction; and Giustolisi and Laucelli (2005), who compared different network optimisation techniques.

Correspondence to: A. Piotrowski
(adampp@igf.edu.pl)

2 Applied models

When dealing with rainfall-runoff modelling, the vector of input variables may consist of IP recent precipitation (x^p) and IR runoff (x^r) observations, i.e. both autoregressive and exogenous inputs are included (Ljung, 1999), together giving the number of input variables (g). The vector of input variables ($\boldsymbol{X}_j$) at the j-th time instant can be presented as

$$\boldsymbol{X}_j = \left(x_j^p, ..., x_{j-IP+1}^p, x_j^r, ..., x_{j-IR+1}^r\right)^T = (x_1,\ x_2,\ ...,\ x_g)^T \quad (1)$$

and corresponds to output variable y_{j+T} – which represents future river runoff. One can then proceed to the stage of determining the prediction model F

$$y_{j+T} = F(\boldsymbol{X}_j) \quad (2)$$

where T is the prediction horizon.

The objective function J to be minimized for each model by proper optimization of parameters (h) is defined in this paper as:

$$J = \min_h \sum_{j=1}^{n} (d_{j+T} - y_{j+T})^2 \quad (3)$$

Published by Copernicus GmbH on behalf of the European Geosciences Union and the American Geophysical Union.

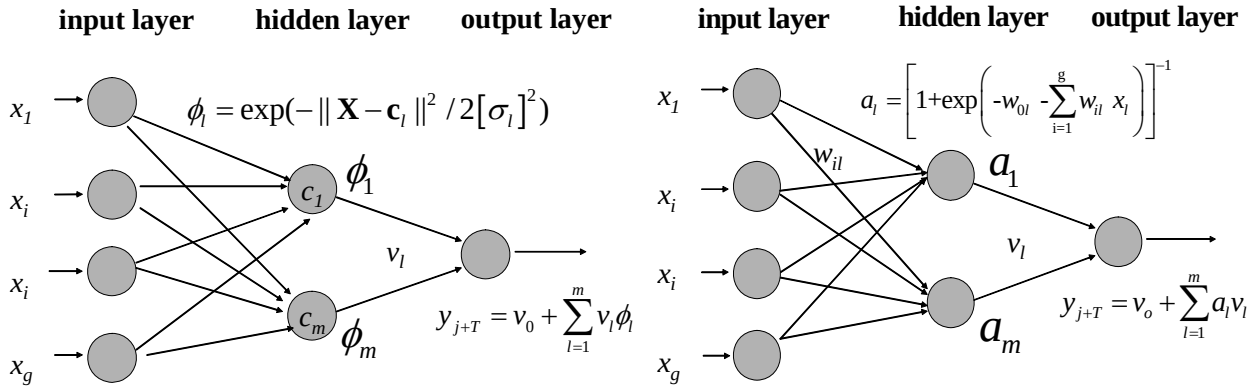

Fig. 1. Radial-Basis Function (left) and Multi-Layer Perceptron (right) neural network.

where n is the number of training or validation set outputs and d_{j+T} is the measured value of flow corresponding to the j-th input vector.

2.1 Linear regression

Linear regression is the well known technique (Box and Jenkins, 1970) enabling one to compute the value of predicted output variable (y_{j+T}) as the weighted sum of g input variables (x_i) and a bias (w_0)

$$y_{j+T} = w_0 + \sum_{i=1}^{g} w_i x_i \tag{4}$$

This approach is obviously not suitable for non-linear relationship modelling, as is necessary in the case of river runoff prediction, and will be used only as a reference model.

2.2 Nearest Neighbours approach (NN)

For the Nearest Neighbours approach, one searches for the K points, representing cases from historical data, that are the most similar (the smallest Euclidean distance in the input space) to the point representing the current situation and applying only these selected parts of the data set for forecasting the unknown output value (Karlsson and Yakowitz, 1987; Shamseldin and O'Connor, 1996; Brath at al., 2002). Data are standardized to the range of [0,1] to obtain the identical scale along each axis in g-dimensional input space. In addition, instead of computing the simple mean of future runoffs of the K data points, it is assumed herein that the function F in Eq. (2) takes the form of a linear regression (Wang et al., 2006), i.e. $g+1$ parameters are to be determined by the least squares method, based on only the K nearest neighbours data. The number of points K, was evaluated by a trial and error procedure, but results differ only slightly for K varying in the range of about 50–100.

2.3 Multi-Layer Perceptron neural networks (MLP)

MLP networks, probably the most popular neural network type in the hydrological sciences, are usually composed of three layers (Fig. 1) comprising several nodes. The number of input and output nodes is equal to the number of input and output variables, but the quantity of hidden nodes is evaluated empirically. The MLP nodes in neighbouring layers are linked via weighted connections.

After being normalized to [0,1], input signals at the j-th time instant from the input nodes are multiplied by proper weights w_{il}, corresponding to connections between input neurons, from which the signal has been dispatched, and neurons in the hidden layer. In each of the m hidden nodes, the weighted sum of all the incoming signals and threshold values (w_{0l}) is computed and then transformed by – in case of this study – the logistic function, giving the value of a_l dispatched by l-th neuron

$$a_l = \left[1+\exp\left(-w_{0l} - \sum_{i=1}^{g} w_{il}\, x_i\right)\right]^{-1} \tag{5}$$

Afterwards, the signals a_l, multiplied by proper weights v_l, are transferred to the neuron of the third layer. In this final stage, the new weighted sum is computed

$$y_{j+T} = v_0 + \sum_{l=1}^{m} a_l v_l \tag{6}$$

and after de-normalization of the output, the sought (forecasted) value y_{j+T} is determined. The values of weights w_{il} and v_l are adaptively modified during the process of training according to the training examples. In the present paper, many models with different numbers of hidden nodes were created and optimized. To avoid stopping in local minima of the objective function, the learning processes used a gradient-based Levenberg-Marquardt optimization algorithm (Haykin, 1994) with the multi start approach from different initial parameter values.

Table 1. Flood periods.

Wave name	Period of data collection	Forecasting period	Number of input vectors	Maximum runoff recorded (m^3/s)
97a	08:00 30 June 1997 – 14:00 13 July 1997	05:00 1 July 1997 – 14:00 13 July 1997	100	1718
97b	20:00 12 July 1997 – 23:00 31 July 1997	17:00 13 July 1997 – 23:00 31 July 1997	147	451
79	08:00 10 June 1979 – 23:00 25 June 1979	05:00 11 June 1979 – 23:00 25 June 1979	119	598
77a	23:00 25 July 1977 – 20:00 10 Aug 1977	20:00 26 July 1977 – 20:00 10 Aug 1977	121	488
77b	11:00 15 Aug 1977 – 20:00 31 Aug 1977	08:00 16 Aug 1977 – 20:00 31 Aug 1977	125	423
65	07:00 23 May 1965 – 22:00 15 June 1965	04:00 24 May 1965 – 22:00 15 June 1965	183	823

Table 2. Correlation coefficients between river runoff and aggregated rainfall with varying time lag (t).

t	0	3	6	9	12	15	18	21	24
$R(y(i): x^p(i-t))$	0,37	0,48	0,57	0,60	0,59	0,56	0,54	0,54	0,53

2.4 Radial-Basis Function neural networks (RBF)

The RBF network, depicted in Fig. 1, differs from the MLP one and includes one hidden layer of special units that preprocess the input and feed a single-layer perceptron. There are different algorithms available when deciding how to optimize the RBF network, and some of them enable each of the three kinds of parameters to be estimated separately.

Each of the m units (the number m is evaluated empirically) in the hidden layer contains centre $\boldsymbol{c}_l$ of the given region of the input space. The corresponding non-linear function φ_l expresses, by means of a distance measure, the similarity between any vector $\boldsymbol{X}$ of input variables and the centre $\boldsymbol{c}_l$. The most commonly adopted Radial-Basis Function is the Gaussian

$$\varphi_l = \exp\left(-\frac{\|\boldsymbol{X} - \boldsymbol{c}_l\|^2}{\sigma}\right) \tag{7}$$

Parameter values $\boldsymbol{c}_l$, σ and v_l (Eq. 9) are to be optimised. The self-organized selection of $\boldsymbol{c}_l$ by the K-means approach is applied in this paper. To obtain the σ parameter value – in this paper assumed to be the same for each $\boldsymbol{c}_l$ – the heuristic rule was applied

$$\sigma = b \cdot \frac{r}{\sqrt{2m}} \tag{8}$$

where r is the maximum value of the Euclidean norm between the two centres, m is the number of centres, and b is an unknown value, which in each case was found empirically. The linear weights v_l in the output layer were optimized by using the pseudoinverse method (Haykin, 1994). The full RBF network gives a model structure with the form

$$y_{j+T} = v_0 + \sum_{l=1}^{m} \varphi_l v_l \tag{9}$$

As in the MLP case, different numbers of hidden units were chosen and optimized a number of times with different initial locations of $\boldsymbol{c}_l$.

3 Data preparation, results and discussion

At the Bardo gauge station, located in the Nysa Kłodzka river, six significant flood periods have been recorded since 1965 (Table 1). The data exploited in this paper consists of runoff measurements, collected every 3 h at the Bardo gauge, and rainfall measurements, aggregated – which reduced significantly the number of parameters to be optimized for each model – by Thiessen polygons method, from 5 stations, located in the Nysa Kłodzka catchment, namely Międzylesie, Kłodzko, Lądek, Słoszów and Mieroszów.

The proper choice of the lag time of runoff and rainfall measurements is very important. Based on the correlation coefficients between rainfall and runoff (Table 2), which show the largest correlation for only 9–12 h and quite slowly diminishes for higher time lags, the maximum lag time of rainfall measurements was limited to 21 h for all versions and

Table 3. Model structure description.

Version	Input nodes	Output nodes	Hidden nodes (MLP)	Centres (RBF)	K-nearest neighbours
V_1-3	10	1	5	20	80
V_2-3	10	1	5	24	80
V_2-6	10	1	7	24	80
V_2-9	10	1	7	18	80

Table 4. Root mean square errors and standard deviation ratio computed for each model case for all versions.

Version	Set	Wave	MLP	RBF	Lin. reg.	NN	MLP	RBF	Lin. reg.	NN
			RMSE				SDR			
V_1-3	training	97a	8,85	14,60	26,67	25,31	0,024	0,040	0,073	0,070
V_1-3	training	97b	8,29	8,99	9,38	7,86	0,074	0,080	0,084	0,070
V_1-3	training	79	5,13	13,14	21,50	30,31	0,053	0,137	0,223	0,315
V_1-3	validation	77a	7,05	6,41	9,01	5,82	0,058	0,053	0,074	0,048
V_1-3	validation	77b	8,57	8,53	10,07	9,23	0,101	0,101	0,119	0,109
V_1-3	test	65	16,84	11,63	13,26	10,11	0,130	0,089	0,102	0,075
V_2-3	test	97a	29,15	15,13	29,44	26,29	0,079	0,042	0,081	0,073
V_2-3	validation	97b	7,89	7,76	7,90	7,16	0,070	0,069	0,071	0,064
V_2-3	validation	79	24,22	24,07	24,29	19,30	0,252	0,251	0,252	0,201
V_2-3	training	77a	5,65	5,59	7,17	6,06	0,046	0,045	0,059	0,050
V_2-3	training	77b	8,04	7,67	8,61	8,46	0,095	0,091	0,102	0,100
V_2-3	training	65	6,90	6,63	10,75	12,56	0,053	0,051	0,083	0,097
V_2-6	test	97a	108,83	30,79	70,36	61,92	0,293	0,084	0,195	0,171
V_2-6	validation	97b	15,37	17,07	16,92	14,44	0,136	0,150	0,151	0,129
V_2-6	validation	79	33,17	45,41	48,41	36,96	0,346	0,473	0,499	0,385
V_2-6	training	77a	9,49	9,99	16,53	12,49	0,077	0,080	0,135	0,102
V_2-6	training	77b	13,53	13,89	17,04	14,46	0,160	0,162	0,201	0,171
V_2-6	training	65	13,16	13,22	22,56	21,17	0,101	0,099	0,174	0,163
V_2-9	test	97a	142,01	71,88	121,28	109,96	0,378	0,188	0,336	0,303
V_2-9	validation	97b	26,47	27,30	24,32	21,46	0,229	0,239	0,217	0,191
V_2-9	validation	79	40,12	58,16	58,26	40,45	0,418	0,606	0,591	0,421
V_2-9	training	77a	13,40	15,97	24,96	17,65	0,106	0,126	0,204	0,143
V_2-9	training	77b	17,58	20,55	24,72	20,09	0,205	0,238	0,291	0,238
V_2-9	training	65	26,62	24,91	38,76	36,75	0,203	0,187	0,298	0,283

lead times. The same number of inputs for each lead time forecast was taken (including at least 3 last runoff measurements).

In a study with a small data set, the way of separating the data into a training set used to evaluate parameter values, a validation set used to determine stopping criteria for the algorithm and to choose the best model of a particular type, and, if possible, a fully optimization-independent test set, may be crucial. In the present paper, it was decided that all observations collected during a particular storm event should belong to the same data set. This way, some flood waves may be considered as independent and become a basis for the appraisal of individual model performance. To perform the forecast in a more convenient way, the largest flood should be made part of the training set. On the other hand, it is especially important to verify the ability of each model to forecast the runoff during the severe 1997 flood (called 97a in Table 1). In the present paper, two versions are analyzed. In version V_1, cases 97a, 97b and 79 compose the training set, 77a and 77b the validation set, and finally 65 – the test set. In version V_2, cases 65, 77a and 77b are included in the training set, 79 and 97b in the validation set, and the highest 97a wave is kept aside as the test set. 3-h lead time forecasts are considered in both versions (named V_1-3 and V_2-3). In the case of version

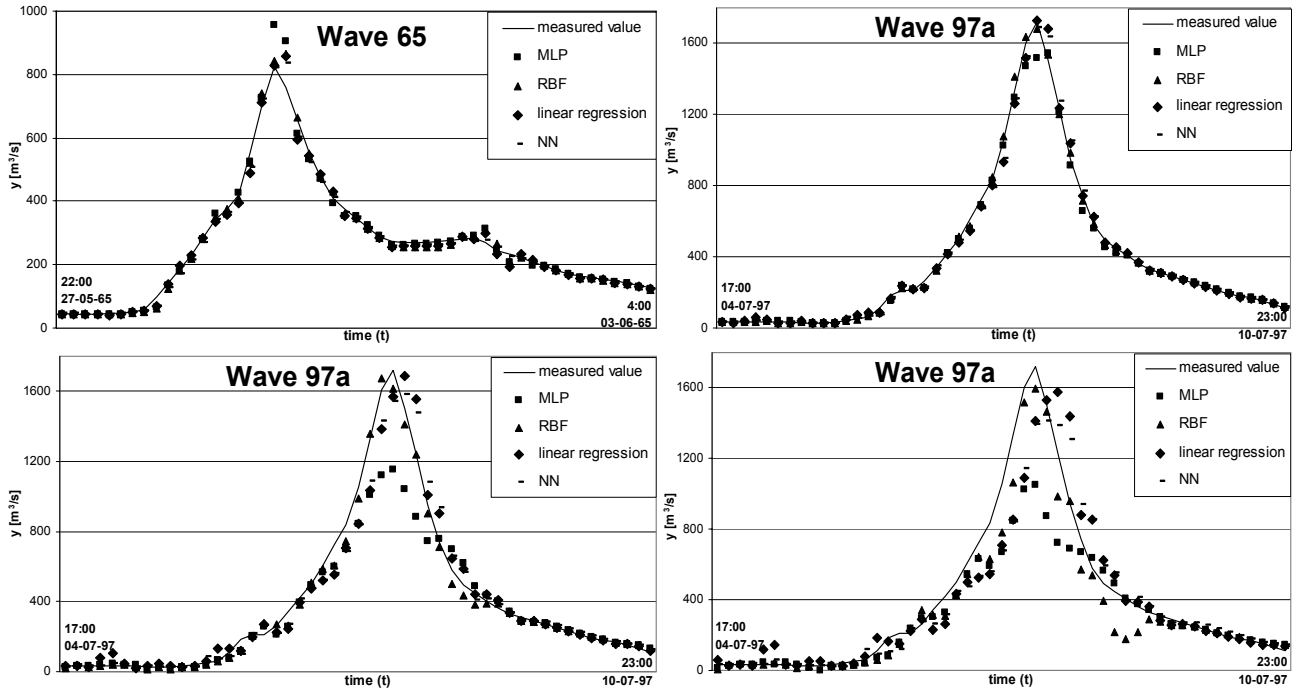

Fig. 2. Test data cases: 3 h lead time flood forecast for 65 (version V_1-3, top left); 3 h (V_2-3, top right), 6 h (V_2-6, bottom left) and 9 h (V_2-9, bottom right) lead time flood forecast for 97a.

V_2, due to its practical importance, also 6- (V_2-6) and 9-h (V_2-9) lead time predictions are made. Altogether, there are four different versions of each model. The final structure of each model version is presented in Table 3.

Three ways of comparing the results will be used: root mean square error (RMSE), computed separately for each wave (n is the number of data points recorded during a storm event)

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_i^f - y_i)^2} \tag{10}$$

standard deviation ratio (SDR), evaluated as

$$\mathrm{SDR} = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\left(y_i^f - y_i\right) - \left(\sum_{j=1}^{n} y_j^f - y_j\right)/n\right)^2}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \left(\sum_{j=1}^{n} y_j\right)/n\right)^2}} \tag{11}$$

and visual comparison.

The obtained results are presented in Table 4 and Fig. 2. In version V_1, only 3-h lead time prediction is performed, showing the surprising result that linear regression turns out to be only slightly inferior to the other methods. Nevertheless, according to the validation and test sets, the RBF and NN approaches are able to forecast runoff slightly better. In Fig. 2, one sees that the poorer performance of the MLP network is due to overestimation of peak flow. The graph confirms that the difference between the other models is negligible.

In version V_2, the forecast for 3- to 9-h lead time are performed. In the case of 3-h forecasts, the 79 storm event is modeled better by the NN approach than by the other methods, whereas the crucial 97a flood forecast obtained from the RBF network significantly outperforms the other techniques. Both RMSE and SDR are almost 50% lower when using this approach. In the 97b case, all of the models perform similarly. With lead times of 6 and 9 h, similar behaviour may be observed: the RBF network outperforms other methods in the case of the 97a storm event, but results obtained for the 97b and 79 events are slightly better if the NN or the MLP approach is applied. It is clear that all of the models only provide reasonable forecasts with a 3-h lead time. The only 6- or 9-h lead time forecasts that are closely related to the measured runoff values are output by the RBF network.

Both the MLP and NN models outperform linear regression for the validation cases. But it is worth noticing that all investigated MLP networks perform significantly worse when applied to the test data sets.

4 Conclusions

In the present paper, Nysa Kłodzka river rainfall-runoff forecasts for flood periods are performed by means of several

data analysis techniques. The obtained results clearly indicate that Radial-Basis Function Neural Networks outperform the other techniques for testing data. If this model had been applied during the most severe 1997 flood, it could have provided useful information about forthcoming events; therefore it can be recommended for practical applications.

Acknowledgements. This work was supported in part by grant 2 P04D 009 29 from the Ministry of Higher Education and Science.

Edited by: A. Witt
Reviewed by: two referees

References

Box, G. E. P. and Jenkins, G. M.: Time Series Analysis: Forecasting and Control, Holden-Day, London, UK, 1970.

Brath, A., Montanari, A., and Toth, E.: Neural networks and non-parametric methods for improving real-time flood forecasting through conceptual hydrological models, Hydrol. Earth Syst. Sci., 6(4), 627–640, 2002.

Dawson, C. W., Harpham, C., Wilby, R. L., and Chen, Y.: Evaluation of artificial neural network techniques for flow forecasting in the River Yangtze, China, Hydrol. Earth Syst. Sci., 6(4), 619–626, 2002.

Giustolisi, O. and Laucelli, D.: Improving generalization of artificial neural networks in rainfall-runoff modelling, Hydrol. Sci. J., 50(3), 439–457, 2005.

Haykin, S.: Neural networks, Macmillan College Publishing Company, Englewood Cliffs, New Jersey, USA, 1994.

Jayawardena, A. W. and Fernando, D. A. K.: Use of radial basis function type artificial neural networks for runoff simulation, Computer-Aided Civil and Infrastructure Engineering, 13, 91–99, 1998.

Karlsson, M. and Yakowitz, S.: Rainfall-runoff forecasting methods, old and new, Stochastic Hydrology and Hydraulics, 1, 303–318, 1987.

Lin, G. F. and Chen, L. H.: A non-linear rainfall-runoff model using radial basis function network, J. Hydrol., 289, 1–8, 2004.

Ljung, L.: System identification. Theory for the user, Prince-Hall PTR, Upper Saddle River, New Jersey, USA, 1999.

Moradkhani, H., Hsu, K., Gupta, H. V., and Sorooshian, S.: Improved streamflow forecasting using self-organizing radial basis function artificial neural networks, J. Hydrol., 295, 246–262, 2004.

Napiórkowski, J. J. (Ed): Modelling and Control of Floods, Publs. Inst. Geophys. Pol. Acad. Sc., E-3(365), 2003.

Napiórkowski, J. J. and Piotrowski, A.: Artificial neural networks as an alternative to the Volterra series in rainfall-runoff modelling, Acta Geophysica Polonica, 53, 459–472, 2005.

Shamseldin, A. Y. and O'Connor, K. M.: A nearest neighbour linear perturbation model for river flow forecasting, J. Hydrol., 179, 353–375, 1996.

Wang, W., Van Gelden, P. H. A. J. M., Vrijling, J. K., and Ma, J.: Forecasting daily streamflow using hybrid ANN models, J. Hydrol., 324, 383–399, 2006.