COMBINING A NEURAL NETWORK AND A NUMERICAL FLOW MODEL IN SYSTEM IDENTIFICATION

István A. Fülöp[1]
Technical University of Budapest, Department of Water Resources Engineering
Mûegyetem rkp 1, Budapest H-1111, Hungary

SUMMARY

Artificial Neural Networks are powerful tools in modeling massively non-linear systems, such as lakes, using a training process based on an adequately large data set. On the other hand, building expertise into a neural network is difficult. The other problem of neural networks is that a neural network trained on the data of one site is not easily adapted to another site, and the applicability is limited if the modeled system considerably changes. In our work we have been examining these problems of the neural networks, trying to overcome the weaknesses with the help of a traditional numerical model.

Keywords: Neural Networks, numerical flow models, generalization, model change, shallow lakes

1 ABOUT NEURAL NETWORKS

A neural network is a non-linear model originally inspired by research of the human nervous system. Current-day neural networks are more often considered to be non-linear statistical methods than the simulations of the natural nervous systems. (Bishop 1995). In our approach a neural network is considered to be a (non-linear) regression model performing a non-linear mapping of the form:

$$ANN(\mathbf{x}) = f(\mathbf{x}, \mathbf{w}) \tag{1}$$

where **x** is the input vector to the neural network, **w** is the vector of adaptive parameters of the neural network, *f* is the non-linear mapping performed by the network (determined by the internal layout of the network) and *ANN* is the output vector produced by the network. **w** is determined from a data set during a process called training. After training, the neural network is used in a process called recall which is to extract the knowledge stored in the adaptive parameters by presenting an input to the network. During recall, the network is presented with previously unseen inputs and is able to generalize, that is provide acceptably good response. (The networks depicted here are called feedforward neural networks having no feedbacks of outputs to the input of the neural network).

The mapping *ANN* is determined by the training set made up of several input-output pairs representing the behavior of the system to be modeled. The training process is

[1] PhD Student

generally a minimization process used to minimize the error between the output of the neural mapping and the output generated by the original system to the same input. The model complexity is generally optimized using the cross-validation well known from other regression methods.

A more detailed introduction to neural networks may be found in (Lippmann 1995), together with the so called Multi-Layer Perceptions we were using in our experiments.

2 EFFECTS OF CHANGES IN THE ORIGINAL SYSTEM

The network performs a mapping converting the input of the network to the desired output of the modeled system. In case of a perfect mapping,

$$ANN(\mathbf{x}) = SYS(\mathbf{x}) \tag{2}$$

where SYS is the output of the original system. Generally this mapping is not perfect, resulting in:

$$ANN(\mathbf{x}) = SYS(\mathbf{x}) + e_1 \tag{3}$$

where ε_1 is the error of the mapping.

The neural network is usually able to perform a reasonably good mapping if the presented input is identical to the **x** vector used in the training process. However, in case of recall, the input usually is a vector not presented to the network during training. Therefore, the mapping tends to be:

$$ANN(\mathbf{z}) = SYS(\mathbf{z}) + e_2 \tag{4}$$

where **z** is a new input not presented during training, and generally $\varepsilon_2 > \varepsilon_1$.

A change in the original system involves a change of the mapping $SYS(\cdot)$ which becomes $SYS_2(\cdot)$, changing the (recall) mapping to:

$$ANN(\mathbf{z}) = SYS_2(\mathbf{z}) + e_3 \tag{5}$$

where generally $\varepsilon_3 > \varepsilon_2 > \varepsilon_1$.

In our experiments we focused on the problem of these changes, using an example of wind induced shallow lake motions.

3 DETAILS OF THE EXPERIMENTS

3.1 The numerical model

In our experiments we used a numerical model called SWAN (Shallow Lake Numerical Model) created at the Technical University of Budapest (SWAN 1995). The model is based on the traditional finite difference solution of the complete shallow water equations and it is equipped with a user friendly interface.

3.2 Neural networks trained on numerical models

Training a neural network on a numerical model has many advantages both from the point of application and the point of research.

The main advantage of application is the considerably faster processing of the neural model compared to that of the traditional numerical model. A neural network applied this way is trained using the relatively slow training process and knowledge is recalled in a fast way giving a quick and reasonably good response in an actual situation. This is a serious advantage in case of high importance systems and critical applications, such as power plants, wherever fast response can be extremely valuable (Enayet Rasul & Paudyal 1994).

An other use of training with numerical models is the ease of testing and analyzing the network (Babovic & Bartoli 1997), since any sort of input-output combinations may be reproduced using the numerical model. This may also lead to new information about the system modeled, as well as the neural network itself.

3.3 Goals of the experiments

In our experiments we focused on the properties and events described by equations 3-4-5, using an example of wind induced motions of model lakes created with the help of the numerical model described above.

We tried to find answers to the following questions:

1. can this approach be used in lake hydrodynamics?
2. what are the effects of changes in the model?
3. what are the results of using a recall input set different from the training set?

3.4 Description of the experiments and results

3.4.1 The posed questions

In our experiments a very simple test basin of dimensions 14 km by 30 km, and a uniform depth of 2 meters was used. To train the neural network, we applied a wind impulse with periodic direction and velocity function.

To recall and test the network, we changed the lake, creating several different peninsulas and measuring the velocities of points uniformly scattered over the lake. The changes in the velocity distribution due to a peninsula can be seen in Figure 1.

The peninsula causes a local non-uniformity in the velocity distribution of the lake. This means, that for any point close to this locality, the transfer function between the wind as input and the water velocity as output changes, as described in eq. 5. If the neural network uses only wind velocities as inputs, this leads to the following mappings:

$$ANN(\mathbf{wind}) = SYSNM(\mathbf{wind}) + \varepsilon_4 \qquad (6),$$

$$SYSNM_2(\mathbf{wind}) \neq SYSNM(\mathbf{wind}) \qquad (7)$$

and

$$ANN(\mathbf{wind}) = SYSNM_2(\mathbf{wind}) + e_2 \qquad (8)$$

where **wind** is the wind input used to train the network using the response of the numerical model *SYSNM*, $SYSNM_2$ is the state (response) of the system to wind input after the change, ε and ε_2 are the errors of the network on the two systems. Since the network inputs (wind velocity) are independent of the batimetry of the lake, the network trained as eq. 6 will not be able to reproduce the response of $SYSNM_2$ because it will not have any information on the change in the batimetry. Therefore $e_2 \gg e_1$

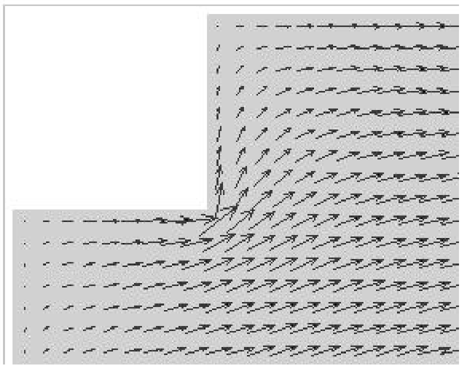

Figure 1. A local disruption in the velocity field caused by a peninsula (the wind impulse is uniform velocity from the West to the East)

If the network is trained as the following:

$$ANN(\mathbf{water}) = SYSNM(\mathbf{water}) + e_8 \qquad (9),$$

where

$$\mathbf{water} = \mathbf{water}(\mathbf{wind}) \qquad (10)$$

is the velocity response of a point in the lake other than the one to be reproduced. In this situation the wind is implicitly represented in the network, and the change in the batimetry is also represented, because it is included in **water(wind)**. The recall will be the following:

$$ANN(\mathbf{water}) = SYSNM_2(\mathbf{water}) + e_8 \qquad (11)$$

However, after the change of the original system, the water - wind function of Equation 9. will be different from the **water**(**wind**) function used to train the network, since the batimetry changes. This implies that the error e_8 will have two components:

1. the component due to the problem explained in eq. 4, that is the difference between the training input data set and the recall input data set

2. the change of relations between the point used as input and the point to be reproduced. This is due to the change of batimetry

The two error components can be separated, as illustrated in the following examples:

Consider a network trained using an arbitrary non-uniform wind input, which is supposed to represent the entire wind induced motion. After the network was trained, it can be recalled using a uniform wind input (for example, a North wind with uniform speed, relaxing in a uniform water velocity) - the error of this recall is completely due to the difference of the training and the recall data set. This problem is also referred to as the location of the generalization points with respect to the *convex hull* of the network (Pelillo 1996). This gives rise to questions concerning the reliability of the network (Bishop 1994).

This problem can also be visualized by creating an auto-associative network (a network trained to reproduced its own input) which performs the following mapping:

$$ANN(\mathbf{water}) = \mathbf{water} + \varepsilon \qquad (12)$$

A linear system would be able to perform this mapping accurately for any **water** values. However, since the neural network used in the experiments is nonlinear, the mapping will only be accurate for the training data set, any other data will be distorted by the difference between the training and the recall input data set.

If the batimetry of the lake is changed and the network is recalled using the same wind input as it was trained from (and the results are compared to the outputs of the lake with the new batimetry), the 2^{nd} type of error is given. This error is approximately identical to the change in the modeled system since the network is reproducing the outputs of the system before the change.

3.5 The experiments

In the experiments, the 2 dimensional flow and wind velocity vectors were represented by their East-West and North-South components.

In the first experiments, the network was given wind inputs to perform a mapping of the form (5). Since the wind input is carrying no information on the batimetry, a network solely relying on implicit wind representation of Equations 8-9 was also used, together with a network getting both implicit and direct wind inputs.

The results confirm that the neural network is able to perform the required mappings accurately using both direct and implicit wind input. The point used as input to the network was a point highly correlated to the one to be reproduced. As the table illustrates, the implicit wind representation is almost as adequate as the direct one. In latter cases, it proves to be even more feasible.

3.6 Effects of model changes

The results of a change in the model are theoretically described in equations 6-8. To see the effect of system changes, the networks described under 3.5 had to be recalled by presenting the same wind input as in the previous case but comparing the response to that of the modified model.

In case of the network relying on the wind inputs only, the results confirm that the network is unable to erform the required mapping, due to the lack of information:

The situation is somewhat different in the next two cases (whenever wind is represented implicitly as well as directly). In the first case, when wind and water velocity are both used as inputs, the results are as follows:

Table 1. Effects of model changes on the ANN with wind and water velocity inputs

Model type	North-South	East-West
Original	0.99	0.99
Peninsula 1	0.97	0.96
Peninsula 2	0.82	0.87
Peninsula 3	0.93	0.94
Peninsula 4	0.65	0.95
Peninsula 5	0.94	0.93

In case of the second experiments, when only water velocities are applied, the results are almost identical to the ones presented above. This indicates that using an implicit representation of the batimetry did improve the modeling capabilities of the network. This improvement can be understood by having a look at the correlation between the point used as the input and the point used as the output of the network:

Table 5. Correlation between the water velocity of the point used as input and the point to be trained

Model type	Correlation
Original	0.96
Peninsula 1	0.91
Peninsula 2	0.84
Peninsula 3	0.83
Peninsula 4	0.92
Peninsula 5	0.87

These data indicate that the improvement is due to the high linear correlation between the two points. It is also obvious from the data, that this improvement is not influenced by the non-relevant direct wind input carrying no additional batimetry information.

The result is limited, however, and can be used when the input-output combinations after the modifications are still remain close to the convex hull of the data set - that is, both two points remain correlated to each other and the inducing wind impulse is almost the same as in case of training. This can be shown by creating an auto-associative network. The network is created using the original model without peninsula and the original periodic wind input. There are, however, four chances of recall, as:

1. original model, original wind
2. original model, modified wind

3. modified model, modified wind
4. modified model, original wind

The results from the previous experiments fall into the 4^{th} class. To see that the improvement is only valid when presenting the original wind to the network, the auto-regressive network has to be recalled using a modified model and modified wind. The modified wind input is a simple East-West unidirectional wind input with uniform velocity. The results are as follows:

Table 3. The results of the neural network are only acceptable when the network is presented with inputs relatively close to the training data set. OM=Original Model, OW = Original Wind, MM = Modified Model, MW = Modified Wind

Model/Wind type	North-South	East-West
OM - OW	0.98	0.98
MM - OW	0.87	0.95
MM - MW	0.34	0.99

the North-South component of the flow is not reproduced properly at all, whilst the East-West component is reproduced with a magnitude bias (not shown in figure), indicating the nonlinear properties of the network.

3.7 Possible applications, further research

The results described in this paper can be used in the online modeling of wind-induced flows of lakes. The fact that batimetry information can be presented to the network by using appropriate inputs is particularly useful because it allows the user to select sites which provide input values and the neural network could be used to calculate the output values of other points. The fact that the points used in training can be selected based on linear combinations is especially useful since it makes an expert’s job easier, because linear correlation is easier to spot than nonlinear similarities.

There are, however, several questions raised, which require further investigations of the area, as:

1. what is the ratio of the error explained by the convex hull of the training data and what is explained by the changes of the modeled system?
2. how large and what type of data set has to be trained to model any possible wind inputs (to the original system)?
3. what is the proper way to represent the batimetry information in the neural network?

Our further research will focus on answering these questions, with the help of the numerical model.

4 CONCLUSIONS

In this paper a neural network application in modeling wind induced shallow lake motions was presented. Our research focused on the problems of changes in the original model.

We found that the neural network is capable to model the outputs of a numerical model with high accuracy. Problems arise, however, when the original system changes. A method is presented to improve the modeling accuracy after system changes and to include batimetry information in the neural network. Limitations of this method are also pointed out, together with the difficulties of applying non-trained data to the neural network.

5 ACKNOWLEDGEMENTS

This work was part of the following projects: „Interface processes and boundary layer dynamics in lake ecosystems” supported by the National Scientific Research Fund of Hungary; Finnish-Hungarian intergovernmental cooperation project on lake research and management; British-Hungarian Science and Technology Cooperation project on „Refined flow and transport modeling in shallow water environment” supported by British Council and the Hungarian Science and Technology Fund.

6 REFERENCES

Babovic, V. & Bartoli, N. 1997. An Application of Artificial Neural Networks in Computational Hydraulics, *Hydroinformatics ’94* pp 233-238 Rotterdam: Balkema

Bishop 1994. Novelty detection and neural networks, *IEE. Proc.-Vis. Image Signal Process.*, Vol 141, No 4., August 1994, pp 217- 222

Bishop, C. M. 1995. *Neural Networks for Pattern Recognition*, Oxford: Clarendon Press

Enayet Rasul, A. Z. - Paudyal, G. N. 1994. Neural Network Application in Hydraulic Structure Operation, *Hydroinformatics ’94* - ed: Verwey-Minns-Baboviæ-Maksimoviæ, pp 233-238, Rotterdam: Balkema

SWAN, 1995.

Pelillo, M. 1996. A Relaxation Algorithm for Estimating the Domain of Validity of Feedforward Neural Networks, *Neural Processing Letters, Volume 3 No 3*. August 1996, Dordrecht: Kluwer Academic Publishers

Lippmann, R. P. An Introduction to Computing with Neural Nets, *IEEE ASSP Magazine*, April 1997, pp 4-22