Nordic Hydrology, 28 (4/5), 1997, 283-296

Neural Networks in the Ice-Correction of Discharge Observations

Paper presented at the Nordic Hydrological Conference
(Akureyri, Iceland – August 1996)

Markus Huttunen and Bertel Vehviläinen
Finnish Environment Institute, Fin-00251 Helsinki

Esko Ukkonen
Dept. of Computer Science, FIN-00014 University of Helsinki

We have applied three models, a neural network, a conceptual model and a combination of these two, a hybrid model, to model the backwater effect of ice in a river. The neural network is a black-box model. It is based mainly on observed data and it lacks the expert knowledge of the system. The conceptual model is based on a physical description of the system. The data is used in optimizing the free parameters of the description. In the hybrid model, the neural network is modified so that the physical description of the conceptual model can be coded into the structure of the network. In the beginning of fitting, the hybrid network already performs as well as the conceptual model. During fitting also the structure of the physical description is optimized, not only the parameters of the description. The three models are rather different in form but in the modeling results there are only slight differences. Mean error of the models in ice-correction is 13-15 m^3/s at an observation station where the mean backwater effect of the ice is 100 m^3/s. The aim of this work is to develop a model for real time estimation of corrected discharge, which is used in error correction of a discharge forecast model. For this purpose the error of the best model is acceptable.

The Problem

River Ice and Water Level

The discharge of a river is usually observed by measuring the water level. The observed water level is transformed to discharge using stage-discharge relation, which is based on a set of discharge measurements. During ice periods the stage-discharge relation may vary since the river ice causes backwater effects on the water level, therefore it can not be used to calculate the discharge.

Modeling Ice-Correction

The aim of this work is to model the water level rise caused by ice. Inputs of the models are daily values of short-wave radiation, observed mean temperature, precipitation and discharge. The wintertime observed discharge is obtained by translating the observed water level into discharge via the summertime and hence erroneous rating curve. The output is an estimate of the correct daily discharge. Another output is ice thickness. There is also observations of ice thickness and it is useful to compare the modeled ice thickness to the observed values.

We have developed three different models to solve this problem. One is a connectionist black-box model, a neural network. The neural network is a network of simple computation units, it is nonlinear and it is based mainly on observed data. The idea of neural modeling was introduced by McCulloch and Pitts (1943), and the nowadays widely used back-propagation training algorithm, which fits the neural network into the observed data, was presented by Rumelhart *et al.* (1986).

The second model is a simple conceptual model, developed by the authors for this application. It is based on observed data and on a simple physical description of the formation and melt of ice cover and its effect on water level.

The third model is a hybrid model, a combination of these two models; the conceptual model is coded into a neural network structure, which is fitted to data using neural network fitting algorithms.

The benefit of the hybrid model is that it utilizes the existing expert knowledge of the system, not only the data as the pure neural network does. Another benefit is that the hybrid model is not restricted to the fixed description as the conceptual problem, but the piecewise linear approximation of the equations of the description is modified automatically on the basis of the data.

Many similar methods to code knowledge into and extract it from a neural network have been presented in the literature. However, the knowledge in our conceptual ice-correction model does differ from these. The knowledge consists of extremely simple difference equations. After fitting the model it is also necessary to get the result back into similar equations to be able to verify consistency of the model. This is why we ended up to develop a new hybrid model, which has the speciality that computation in the network is very similar to the conceptual model. It is not necessary to extract rules explicitly from the network after fitting because the network itself is understandable.

Usage of the Ice-Correction in the Watershed Model

A watershed model simulates discharge using precipitation, temperature and potential evaporation as inputs. Such watershed model is fitted to a long period of observed data, usually from 10 to 20 years. When a new period of data is simulated with the fitted model, usually the observed and simulated discharges do not match exactly. In real time models most of the error between the observed and simulated values is caused by missing temperature and precipitation observations. The missing values are approximated from the observations of real time weather stations, which are located far from each other. In the case of errornous inputs, the most appropriate way to correct the error of the model is to make slight corrections to the inputs.

One of the most important applications of the watershed models is to forecast discharge and water level. In forecasting the benefit of the simulation correction is that a more accurate state of the model is obtained in the beginning of the forecast and that way the forecasts are more accurate. An automatic algorithm for the correction procedure is described in Kuha (1993) and Vehviläinen (1994). This correction procedure is also denoted data assimilation, as for example in Chui and Chen (1991).

The ice-correction model gives an estimate of the ice-corrected discharge, which is then used in correction of the watershed model. The aim is that neither of the models needs manual control and so the whole procedure can be made automatically. First the ice-correction model corrects the discharge observation and then the watershed simulation is corrected to match the corrected observation. This procedure also verifies the results of the ice-correction model. If it is not possible to correct the watershed model to simulate the ice-corrected discharge, it is obvious that the ice-correction is incorrect.

Connectionist Black-Box Model

Computation Units

The connectionist network consists of nodes and connections between the nodes, as in Fig. 1 and Fig. 2. Each node computes a simple function of its input, for example hyperbolic tangent. Each connection has a weight and direction associated with itself, from one node to another. The value of a connection is the product of its weight and the value of its starting node. The input of a node is the sum of the values of the connections ending to it. The value of a node is usually hyperbolic tangent of its input, Fig. 1. Nodes are located in layers as in Fig. 2. All connections ending to a node begin from nodes in the previous layer. All connections beginning from a node end to nodes in the next layer. This is the basic structure of a feedforward neural network. A more complete introduction to this topic can be found for example in Herz *et al.* (1991).

In our case we have a few recurrent connections in the network. The idea of these is to feed the current internal state of the system to the next step of the simulated time. The output from the previous time step could simply be fed to the next time step. But the disadvantage of this method is that output of the model is not necessarily a good description of the state of the system. Therefore we allow the backward connections to start from defined nodes in the network. While the model is fitted to the observed data the fitting algorithm is allowed to select what information to feed to the next time step.

The first layer in the network is referred to as the input layer and the last one as

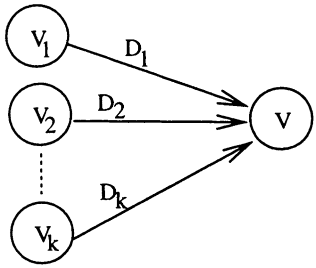

Fig. 1. The neural network consists of nodes and connections. Each connection has weight (*D*). The value of a node (*V*) is hyperbolic tangent of the sum of its inputs (*V*=tanh ($\Sigma^{k}_{i=1} d_i v_i$)).

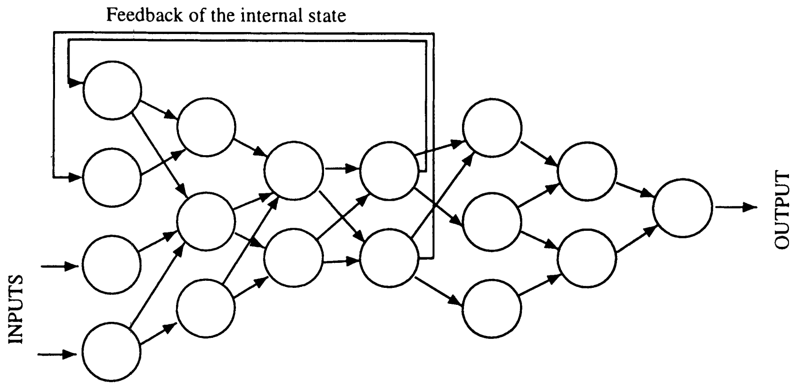

Fig. 2. An example of a network. The value of each node is updated once in each time step. Backward connections feed the state of the system to the next time step.

the output layer. At each time step the inputs of the model are assigned to values of specified input nodes in the input layer. The values for the rest of the nodes in the input layer come from the previous time step through the backward connections (Fig. 2). After the values of the nodes in the input layer have beed assigned, the values of the nodes in the second layer are computed and after this the values of the nodes in the third layer, *etc*. The output of the model is the same as the value of the output node, the one node in the output layer. After the output of the time step is read, the next time step is computed in the same way. The inputs of the time step are assigned to the input nodes, values of all the nodes are computed and the output is read from the output node.

We have used observation data from one hydrological observation station. Daily observations are from the years 1961-93. Ice-thickness and ice-correction are modeled using separate networks. Both modes have three inputs. The inputs of the ice-thickness model are temperature, precipitation and short-wave radiation and those of the ice-correction model are precipitation, modeled ice-thickness and observed water level. In our application we found out by experimental tests that the number of the nodes should be 30-50 and the number of the layers seven, as in Fig. 2.

The error function, given by Eq. (1), is in our case the weighted sum of errors from the data and from a set of boundary conditions. The boundary conditions are constraining rules that are set by an expert. The purpose of the boundary conditions is to reject models which fit well on the data but in other respects do not correspond to the existing knowledge of the system. There are coefficients c_i for errors in each dataset and in each boundary condition.

$$\text{Error} = \sum_{i=1}^{2} c_i \,\text{DataSetError}_i + \sum_{i=1}^{n} c_{i+2}\,\text{BoundaryConditionError}_i \quad (1)$$

Evolutionary Fitting

The neural network model has a number of free parameters. The parameters are the number of nodes in each layer, the existence of connections between the nodes and the weights of the connections. The model is fitted to the observation data by optimizing the free parameters. This is carried out by minimizing the error function (Eq. (1)) using an evolutionary optimization algorithm.

The basic idea of an evolutionary algorithm is to optimize simultaneously a set of parameter combinations, where each parameter combination defines one model. We call this set of parameter combinations (*i.e.*, models) a working set. A new parameter combination is created by applying a so called cross-over procedure. Two parent combinations in the working set are selected randomly and values for each parameter in the new combination are picked randomly from one or the another parent combination. After this small random modifications, commonly referred to as mutations are made to the parameter values in the new combination and the resulting combination is added to the working set. The set must contain a constant number of combinations, so after adding there a new combination, the worst or randomly one of the worst combinations is removed.

An evolutionary optimization algorithm is robust; it can avoid local minimas of the error function, at least to some extent. Unfortunately the algorithm is also time demanding, computation time can be long, even days or weeks. A more complete introduction to the evolutionary optimization can be found in Fogel and Fogel (1994) or Fogel (1995).

In our case the working set consists of a number of different networks. Initially one network was created by finding manually an approximate number of nodes. Networks with different number of nodes were tested by fitting the weights of the connections into the data and testing how well the resulting network forecasts in new data. The best one of the tested networks was then copied a number of times to create the members to the initial working set. Small random changes were made to the structures and weights of each copy. The resulting set of networks was used as the initial working set to the evolutionary optimization.

The optimization algorithm creates a new network by combining two networks in the working set. The two networks are chosen randomly, networks are cut into two pieces by picking randomly a layer and a node where to cut. The new network is a combination of the first part of the one network and the second part of the other network. The new network is modified randomly by adding and deleting nodes and connections and changing weights of the connections. After this the weights are optimized using Hooke-Jeeves direct search optimization algorithm. The Hooke-Jeeves algorithm was originally presented by Hooke and Jeeves (1961) and time complexity properties of the algorithm (*i.e.* how the running time of the algorithm depends on the number of the optimized parameters) was recently presented by Kuha (1997). After the direct search optimization the resulting network is added to the working set and the worst network in the set is removed. The direct search optimization is based on comparing the values of the error function when the values of the parameters are changed by small steps. The gradient of the error function is not needed and hence the method can easily be used with an error function containing boundary conditions. When used with evolutionary optimization, the direct search optimization makes the convergence faster in the beginning of optimization.

The data set is divided into two sets, fitting data and validation data. The data is in one year periods, each period begins from 1st of August and ends at 31st of July. Randomly 50% of the years are chosen into the fitting data set and the rest are used in validation. The direct search algorithm optimizes the weights by comparing the model performance in the fitting data. After the direct search optimization the performance of the model in the validation data is tested and based on this it is decided which network in the working set to delete. The aim of the validation is to have an optimal number of nodes and connections in the network. If there is too many nodes and connections, it is obvious that the direct search algorithm overfits the network into the fitting data and performance in the validation data is poor. On the other hand if there is not enough nodes and connections, the direct search algorithm is not able to fit the network well enough and the performance is poor as well.

The evolutionary and the direct search algorithms are not the only possibilities in training neural networks. The more commonly used algorithm is the back-propagation algorithm, see for example Rumelhart *et al.* (1986) or Herz *et al.* (1991). It is a gradient method and works by propagating the error of the network backward from the output node. The back-propagation algorithm is an order of magnitude faster than the evolutionary or the direct search algorithm, because the gradient tells how to change all the parameters at the same time. However, the back-propagation algorithm is not suitable for optimizing the structure of the network, but only the values of the weights. The back-propagation algorithm could be applied instead of the direct search algorithm. However, the direct search algorithm is simpler to apply when the error function (Eq. (1)) contains a set of boundary conditions and it is especially simpler to apply when the connection functions are not linear but piecewise linear as in the case of the hybrid model.

Conceptual Model

The conceptual ice-correction model is based on a simple description of the formation, melt and backwater effect of the ice cover. Ice formation and melt depends on the amount of snow on ice, ice-thickness and equilibrium temperature, which depends on daily mean temperature and short-wave radiation. Backwater effect depends on ice-thickness and observed discharge. Equilibrium temperature T_{eq} defines the point when ice is not melt nor new ice is formating. The idea of equilibrium temperature is from Bengtsson (1976), here it is used as in Eq. (2)

$$T_{eq} = T + \frac{\text{ShortWaveRadiation} - \text{parameter}_1}{\text{parameter}_2} - \text{parameter}_3 \qquad (2)$$

Degree-day and relative ice cover area ($0 \leq \text{Open} \leq 1$) are defined in Eqs. (3) and (5). Indexes $t-1$ and t mean consecutive time steps. There are also parameters for upper and lower limit of degree-day, see Eq. (4)

$$\text{TemperatureSum}_t = \text{TemperatureSum}_{t-1} + T_{eq} \qquad (3)$$

$$\text{parameter}_4 \leq \text{TemperatureSum}_t \leq \text{parameter}_5 \qquad (4)$$

$$\text{Open}_t = \text{Open}_{t-1} + \text{parameter}_6 \times \text{TemperatureSum}_t \qquad (5)$$

Ice formation and melt is defined as in Eq. (8). The values of the parameters in Eqs. (6) and (7) depends on if T_{eq} is greater than or less than zero. Finally the ice-correction depends on the observed discharge and ice-thickness as in Eq. (9)

$$\text{COEFF}_{ice} = \frac{1}{1+\text{Icethickness} \times \text{parameter}} \qquad (6)$$

$$\text{COEFF}_{snow} = \frac{1}{1+\text{PrecipitationOnIce} \times \text{parameter}} \qquad (7)$$

$$\text{Icethickness}_t = \text{Icethickness}_{t-1} + \text{parameter}_9 \times T_{eq} \times \text{Open} + \text{parameter}_{10} \times T_{eq} \text{ (1-Open) } \text{COEFF}_{ice} \times \text{COEFF}_{snow} \qquad (8)$$

$$\text{Corrected Discharge} = \text{ObservedDischarge-param}_{11} \times \text{Icethickness}^{\text{param}_{12}} \times \max(0, \text{Observed Discharge-param}_{13})^{\text{param}_{14}} \qquad (9)$$

We fitted the parameters of the model, $parameter_{1,\ldots,14}$ to the data using the Hooke-Jeeves direct search algorithm. The initial values for the parameters were set by an expert that knows the ice process. The same fitting and validation data sets were used as for the neural network. The values of the parameters were optimized iteratively by modifying the values of the parameters slightly and accepting modifications which improve the performance of the model in the fitting data. The fitting was continued until the performance of the model in the validation data set was not improved. The error function, which defines the goodness of a parameter value combination was also the same as for the neural network, Eq. (1).

Hybrid Model

The neural network model does not utilize the already existing knowledge about the ice-correction model. A conceptual model does utilize the knowledge but only the parameters of the equations are fitted to data. During fitting the conceptual model, it might occur that the equations of the model should be modified somehow, in order to get the model fitted better in data. Usually in this case the fitting is stopped, the modifications are made manually and then the fitting is continued. However, this may be a long iterative process. The aim of the hybrid model is that it makes automatically some of this manual work suggesting useful modifications into the structure of the model, which an expert can utilize in improving the conceptual description. Our hybrid model offers a method that is able on the one hand to utilize the knowledge as in the conceptual model and on the other hand is able to change the coded knowledge automatically by optimizing the structure of the model using an evolutionary algorithm.

Similiar hybrid methods to code knowledge into and extract it from a neural network have been presented in the literature. In Towell (1991) and in Towell and Shavlik (1994) a method is described for inserting and extracting symbolic knowledge. In Opiz and Shavlik (1997) a genetic algorithm is presented for optimization of neural networks. The most similar application is in Cozzio-Büeler (1995) where differential equations are coded into a neural network. However, the knowledge in these applications is not similiar to the knowledge in our conceptual ice-correction model. Therefore we ended up to develop a new hybrid model, in which the representation of the knowledge of our case is compact and easy to understand.

We modified the neural network so that the resulting hybrid model can easily utilize the equations of the conceptual model. In the hybrid network, the value of a connection is piecewise linear function of the value of the start node, as in Fig. 3. A connection can end to another connection instead of a node, in this case the value of the connection is a multiplicative coefficient to the value of the target connection, as in Fig. 2. Each node contains a linear activation function. This hybrid model can also be made probabilistic by defining distribution for the values of the connection function, as in Fig. 3. Here the distribution is uniform and distribution parameters are optimized to get confidence limits as narrow as possible (Fig. 6).

The variables of a conceptual model are the inputs and the outputs of the model and the intermediate results defined by the equations. Hence the variables are the same as the left hand sides of the equations of the conceptual model. Into the hybrid network can be coded all functions of one variable (like $a = f(b)$). With two or more

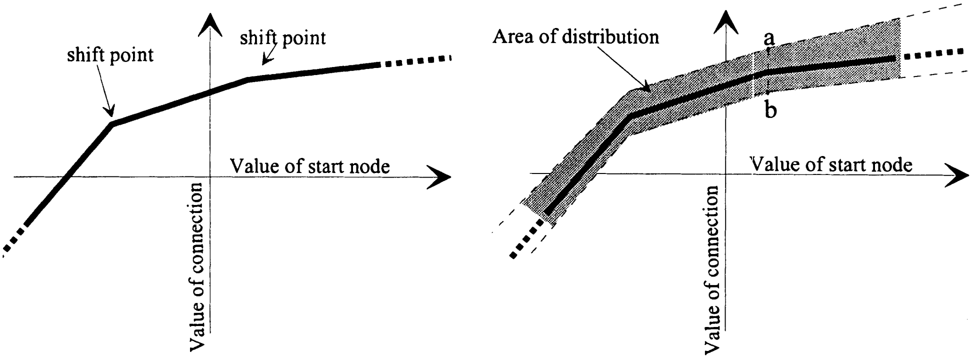

Fig. 3. (Left) A connection in the network of hybrid model is a piecewise linear function. (Right) The model is made probabilistic by adding probability distribution to the values of the connection function. The width of this uniform distribution is defined piecewise lineary.

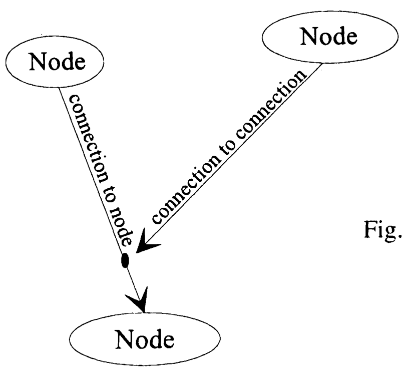

Fig. 4. A connection of the hybrid network can end to another connection. In this case the value of the connection is used as a coefficient to the value of the target connection.

variables only the basic arithmetic operations, *i.e.* addition, subtraction, multiplication and division are available ($a = f(b,c) = b + c$). This means that equations of a conceptual model have to be converted so that the right hand side of the equations contains either only one variable (and the function to be applied on it) or only basic arithmetic operations. The hybrid network has one node for each variable of the conceptual model. The nodes in Fig. 5 are the variables of the conceptual ice-correction model. An exception is that complicated equations may have to be coded using additional nodes for intermediate results. Another exception is that for some variables one needs only a connection, for example Eqs. (6) and (7).

Equations that contain a function of one variable are represented by one connection, which is simply a piecewise linear approximation of the function. If the equation contains sum of two or more variables as in Eq. (2), then one needs as many connections as there are variables in the sum. For example Eq. (2) is coded by con-

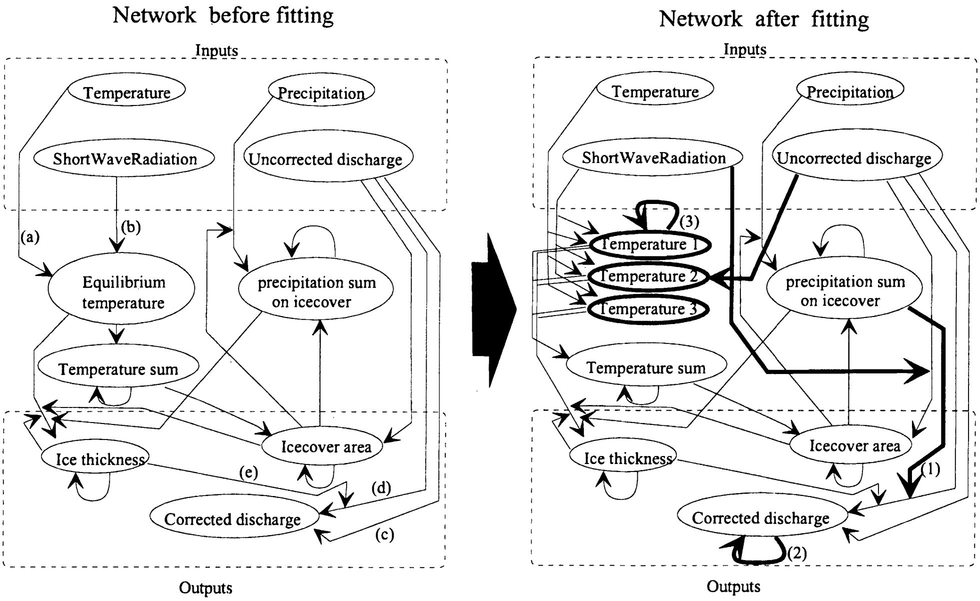

Fig. 5. Nodes and connections in the conceptual model (network before fitting) and hybrid model (network after fitting). Thick line marks nodes and connections that were added by the fitting algorithm.

nections (a) and (b) in Fig. 5. Connections ending to another connection are needed when the equation contains a product of variables. One of such variables is coded as a connection ending to the target node and the other parts of the product are coded as connections ending to that connection. For example, in the coding of Eq. (9) in Fig. 5 there are connections (c) and (d) for the sum and connection (e) for the product. Before starting the evolutionary algorithm, that modifies the structure of the network the hybrid model is equivalent to the conceptual model, except for the error that comes from the piecewise linear approximation of functions.

It is a known fact that a piecewise linear function can approximate any function with arbitrarily small error. An expert is needed who knows the conceptual model and is able to define the amount of acceptable approximation error for each function. The number and placement of shift points in the piecewise linear connection functions are initialy set based on the original equations and this definition of acceptable error.

The parameters of the hybrid model consist of the number of nodes, the existence of connections between the nodes and the number and placement of shift points in the piecewise linear functions of the connections. If the model is probabilistic then there is also parameters for the width of distributions. Width is defined by two parameters at each shift point (Fig. 3). Parameters are fitted using the same evolutionary optimization algorithm as in fitting the neural network ice-correction model.

Also the validation process is the same as in fitting the neural network. The placement of the shift points are fitted by the direct search algorithm into the fitting data, which is 50% of the entire data set. The rest 50% of the data is the validation set, which is used in testing how well the fitted model forecasts new data.

An example of a network after fitting is in Fig. 5. The evolutionary algorithm has added some meaningful nodes and connections to the network; connection (1) in Fig. 5 means that the amount of ice-correction depends on the precipitation sum on the ice-cover. Connection (2) means that the corrected discharge depends on the corrected discharge at the previous day and in (3) the new node and connection means that the equilibrium temperature depends on the equilibrium temperature at the previous day.

Results

We have applied these models so far to one observation station, which is Tornionjoki Karunki in the Northern Finland. Observation data is from the years 1961-93, an example of the data is in Fig. 6. The models are fitted to two datasets, observed ice-thickness and manual graphical ice-correction. The manual graphical ice-correction, as described in Leppäjärvi and Vehviläinen (1994) and Hyvärinen (1986), is done manually by an expert. It is based on observations of temperature, precipitation and incorrected discharge, discharges of near-by stations and usually on one or two winter-time discharge measurements. Manual ice-correction is laborious and that is why it is not suitable for real time ice-correction, which was the aim of this work. However, while making the ice-correction the expert has taken into consideration all significant factors as well as one can, and that is why it is the best available data for fitting the models.

Our error function (Eq. (1)) consists of errors from both data sets and from a set of boundary conditions. The boundary conditions are set to reject some models that fit well on the data but in other respects do not correspond to the existing knowledge of the system. The data sets are divided into two equally sized parts into fitting and validation sets.

Mean discharge of the Tornionjoki Karunki station between the 1st of October and the 30th of June is 369 m^3/s and mean effect of the ice is 100 m^3/s (Fig. 6). Mean error in the ice-corrected discharges of the models is 13.0-15.5 m^3/s (Table 1). Mean error in ice-thickness is 5.4-9.5 *cm*. The neural network model was fitted also for modeling ice breakup day and the resulting mean error was 1.4 days.

In ice-thickness and ice-correction modeling the best model was the hybrid model. In Table 1 'before structure fitting' means performance of the model before the evolutionary algorithm had been applied. At that point in the neural network only the weights of the connections had been fitted and in the hybrid model only the positions of the shift points of the connection functions had been fitted. Before struc-

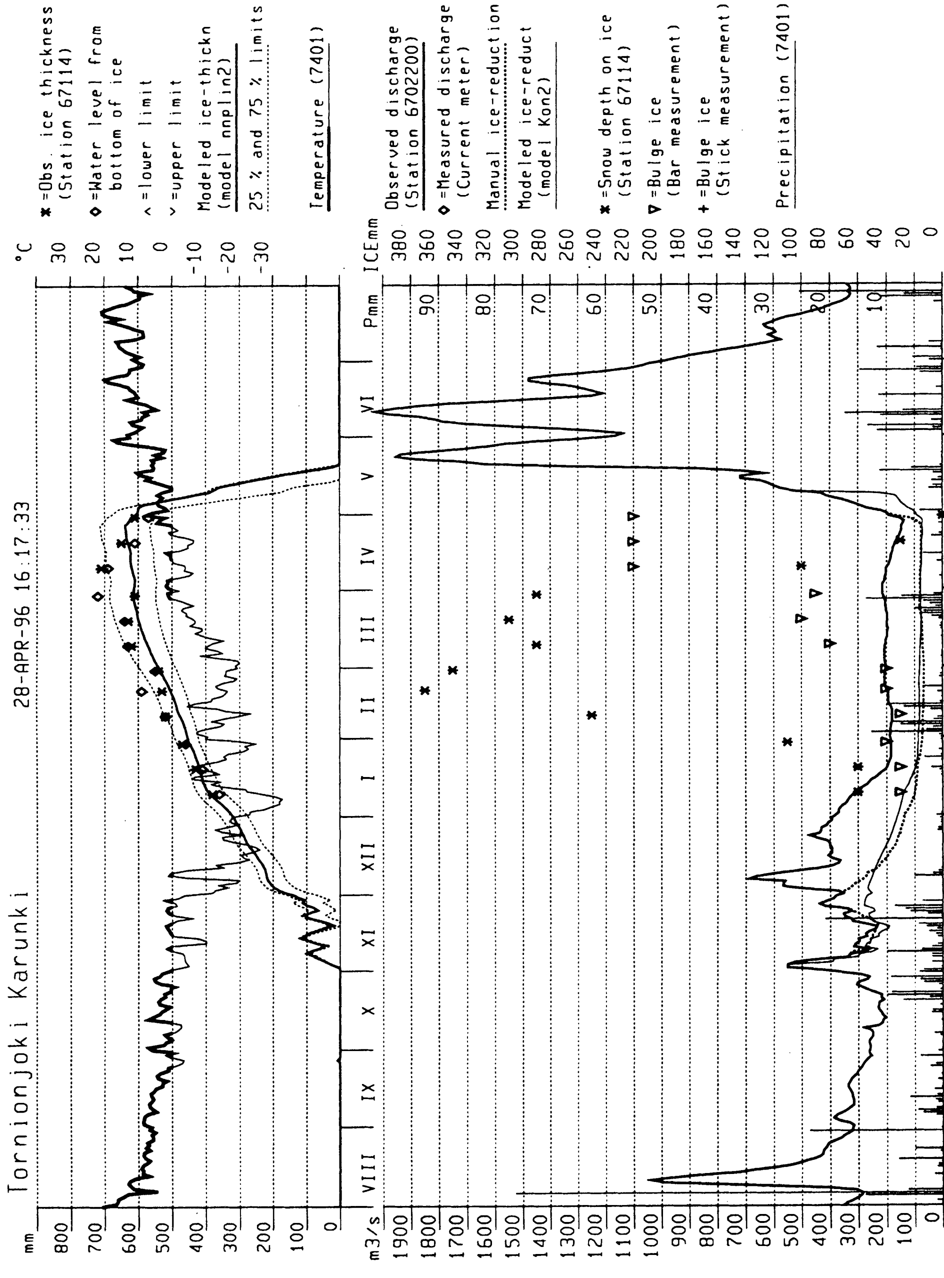

Fig. 6. Observed and modeled (hybrid model) values for ice-thickness and ice-correction at the observation station Tornionjoki Karunki for the winter 1986-87. The ice-thickness model is probabilistic, with confidence limits within which at least 50% of the observations lay. Observation data used in training the models are from the years 1961-93.

Table 1 – The difference between the observed and the modeled values at the observation station Tornionjoki Karunki. The error of the discharge is calculated between the 1st of October and the 30th of June. The observation data is from the years 1961-93. The shown errors are the errors in the validation data, which have not been used in fitting the parameters.

Model	Validation error Ice-thickness cm	Ice breakup days	Discharge m^3/s	Computation time in fitting hours
Neural network				
before structure fitting	10.2	-	18.0	< 1
after structure fitting	7.2	1.4	15.5	1000
Conceptual model	9.5	-	13.3	< 1
Hybrid model				
before structure fitting	9.0	-	13.3	< 1
after structure fitting	5.4	-	13.0	500

ture fitting the neural network model performs in ice-correction considerably worse than the conceptual model and after the time-consuming structure fitting it still performs 15% worse than the conceptual model. The hybrid model performs before structure fitting as well or slightly better than the conceptual model. After the structure fitting it performs in ice-correction 2% better and in ice-thickness considerably better than the conceptual model. The modification that the evolutionary algorithm has made in the structure of the hybrid model can be verified (Fig. 5). The modifications are meaningful and useful when when the conceptual description is developed further.

Acknowledgements

The financial support from the IHP Committee of the Academy of Finland (grant no. 8521) and from Imatran Voiman Säätiö foundation is gratefully acknowledged. We would also like to express our gratitude to Ms. Raija Leppäjärvi and Mr. Matti Ekholm for their help in understanding the data and support during the work.

References

Bengtsson, L. (1976) Snowmelt Estimated from Energy Budget Studies, *Nordic Hydrology, Vol. 7*, pp. 3-18.

Chui, D. K., and Chen, G. (1991) Kalman filter with real-time applications, *Springer Series in Information Sciences, Vol. 17*, Springer-Verlag.

Cozzio-Büeler, R. A. (1995) The Design of Neural Networks using a priori Knowledge, A dissertation, Swiss Federal Institute of Technology.

Fogel, D. B., and Fogel, L. J. (Eds.) (1994) *IEEE Transactions on Neural Networks*: Special Issue on Evolutionary Programming, *Vol. 5*, No. 1, January.

Fogel, D. B. (1995) *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, Piscataway, NJ: IEEE Press.

Herz, J., Krogh, A., and Palmer, R. G. (1991) *Introduction to The Theory of Neural Computation*, Santa Fe Institute stüdies in the sciences of complexity, Addison Wesley 1991.

Hooke, R., and Jeeves, T. A. (1961) "Direct Search" Solution of Numerical and Statistical Problems, *Journal of the Association for Computing Machinery, Vol. 8.*

Hyvärinen, V. (1986) Some practical aspects of graphical ice reduction, Proceedings of Sixth International Northern Research Basins Symposium/Workshop (Field Measurements Under Winter Conditions). MTU, Hougton, Michigan, Jan. 26-30, pp. 25-29.

Kuha, J. (1993) Funktion arvojen vertailuun perustuva epälineaarinen optimointi, M.Sc. Thesis (in Finnish), University of Helsinki, Department of Computer Science.

Kuha, J. (1997) Linearity in time complexity of Hooke-Jeeves algorithm, Preprint, Nokia Research Center, Finland.

Leppäjärvi, R., and Vehviläinen, B. (1994) Ice reduction of winter discharge by watershed models, Publications of the Water and Environment Research Institute, Finland, No. 17.

McCulloch, W. S., and Pitts, W. (1943) A Logical Calculus of Ideas Immanent in Nervous Activity, *Bulletin of Mathematical Biophysics, Vol. 5*, pp. 115-133.

Opiz, D. W., and Shavlik, J. W. (1997) Connectionist Theory Refinement: Genetically Searching the Space of Network Topologies, *Journal of Artifirial Intelligence Research, Vol. 6*, pp. 177-209.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986) Learning Representations by Back-Propagating Errors, *Nature, Vol. 323*, pp. 533-536.

Towell, G. G. (1991) Symbolic Knowledge and Neural Networks: Insertion, Refinement and Extraction, Ph.D Thesis, University of Wisconsin – Madison.

Towell, G. G., and Shavlik, J. W. (1994) Knowledge-based artifical neural networks, *Artificial Intelligence, Vol. 70*, pp. 119-165.

Vehviläinen, B. (1994) The watershed simulation and forecasting system in the National Board of Waters and the Environment, Publications of the Water and Environment Research Institute, Finland, No. 17.

Received: 15 October, 1996
Revised: 22 August, 1997
Accepted: 16 September, 1997

Address:

Markus Huttunen, Bertel Vehviläinen,
Finnish Environment Institute,
P.O.Box 140,
FIN-00251 Helsinki,
Finland.

Email: Markus.Huttunen@Vyh.Fi
Bertel.Vehvilainen@Vyh.Fi

Esko Ukkonen,
Department of Computer Science,
University of Helsinki,
PO Box 26 (Teollisuuskatu 23),
FIN-00014 University of Helsinki,
Finland.

Email: ukkonen@cs.Helsinki.FI