Berthold, T.; Milbradt, P.; Berkhahn, V.

Determination of Network Topology for Ann-Bathymetric Models

Proceedings of ninth International Conference on Hydro-Science and Engineering (ICHE 2010), IIT Madras, Chennai, India.
2 - 5 August 2010

DETERMINATION OF NETWORK TOPOLOGY FOR ANN-BATHYMETRIC MODELS

T. Berthold[1], P. Milbradt[2] and V. Berkhahn[3]

Abstract*: Information on bathymetries and how they change over time forms the basis of simulation- and prediction-models for tidal currents and waves or morphodynamics. The most important value is the depth of the sea with respect to position and time. Bathymetric models are typically based on discrete measured data combined with an interpretation method.*
An alternative approach is to train an artificial neural network (ANN) using the measured data h = (x,y,t) as training patterns. An ANN is a data-based model that is able to "learn" a mapping represented by a discrete set of data. After a training process, the ANN approximates the implicitly given mapping. A multilayer perceptron is used to act as an ANN-bathymetric model, which was trained by the well-known backpropagation learning rule.
During a trial and error process of choosing an appropriate network topology and training sets, good approximations of the seabed could be achieved for both time independent and time dependent models. The different results, depending mainly on the network topology, led to the question: Which topology is sufficient to represent a given bathymetry well and how are characteristic structures represented by a certain topology? This paper presents some rules on how to choose an appropriate network topology regarding a priori knowledge of the seabed.
Several examples will demonstrate the possibilities and limits of the ANN-bathymetric model.

Keywords*: ANN; network topology; bathymetric model; representability.*

INTRODUCTION

Information on bathymetries and how they change over time forms the basis of simulation- and prediction-models for tidal currents and waves or morphodynamics. The most important value is the depth of the sea with respect to position and time. Information is usually gained by measuring a domain of interest. For a certain location (x,y) at a given point in time (t) the depth of the sea (h) is measured. The established measurement techniques provide exactly one value for a given position, so that the relation between the depth and the position in space and time can be formulated as a function

1 Research assistant, Institute for Computer Science in Civil Engineering, Leibniz University of Hanover, Callinstraße 34, 30167 Hanover, Germany, Email: berthold@bauinf.uni-hannover.de
2 PD Dr.-Ing., smile consult GmbH, Vahrenwalder Straße 7, 30165 Hannover, Germany, Email: milbradt@smileconsult.de
3 PD Dr.-Ing., Institute for Computer Science in Civil Engineering, Leibniz University of Hanover, Callinstraße 34, 30167 Hanover, Germany, Email: berkhahn@bauinf.uni-hannover.de

$$h = f(x, y, t) \quad (1)$$

Furthermore, the measurements are only taken at discrete points in space and time. The use of an interpretation method leads to a continuous description of the bathymetry. Bathymetric models are typically based on a set of discrete measured data combined with an interpretation method.

An alternative approach is to train an artificial neural network (ANN) using the measured data $h = f(x, y, t)$ as training patterns as described in Berthold and Milbradt (2009). An ANN is a data-based model that is able to "learn" a mapping represented by a discrete set of data. After a training process, the ANN approximates the implicitly given mapping. The next section will give a short overview of ANN. The settings of the model will be presented which will be the basis for the investigations.

During a trial and error process of choosing appropriate network topologies and training sets, good approximations of the seabed could be achieved for both time independent and time dependent models. The different results, depending mainly on the network topology, led to the question: Which topology is sufficient to represent a given bathymetry well and how are characteristic structures represented by a certain topology? These questions are the main goal of this paper and are dealt with in the third section.

ANN-BATHYMETRIC MODEL

A bathymetric model is typically based on measured data, which is obtained by an echo sounder for example. The echo sounder measures the depth at certain locations, which leads to a finite set of data. Assuming that there is only one measured value for each measuring location, the depth h can be interpreted as a function that depends on the location: $h = f(x, y)$. To get a continuous surface from the discrete distribution, the model has to provide an interpretation method. The interpretation method specifies how to calculate the value h for a given location (x, y). Interpretation methods can be divided into interpolation and approximation methods. An interpolation method I has to hit the measured values h_i at the related position p_i exactly:

$$I(p_i) = h_i \quad (2)$$

To this, an approximation is a more general method, where the goal is to find a function A that nearly hits the measured data, but does not have to hit exactly:

$$A(p_i) \approx h_i \quad (3)$$

In this paper an ANN is used to serve as a bathymetric model.

ANN: theoretical overview

In the context of ANN, the functioning of the human brain and its ability to abstract is tried to be carried to the computer as a simplified model. As the information in the brain is mainly processed and influenced by neurons and synapses, these build the basic elements of ANN as well. A neuron can be understood as a ”mini-processor” that produces an output signal, if the incoming signal exceeds a certain level. This output is sent to connected neurons via the nerve tracts. Each neuron is connected to its predecessor and successor neurons through other cells, which are called synapses. A synapse influences the magnitude of the transmitted signal by decreasing or increasing it. That has an important effect on the signal processing and therefore establishes the basis for many learning algorithms. A synapse is mostly modeled as a scalar value that represents the weight the signal is multiplied with.

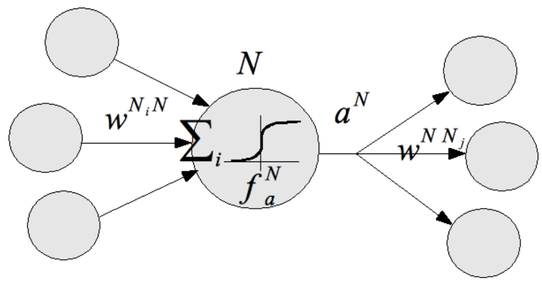

Fig. 1. Signal processing in an ANN: Each neuron N produces an output signal depending on the incoming signals and sends it to its successors. The variable weights of the synapses influence the signals.

Special neurons act as input and output neurons. For given input values information is propagated through the network of neurons by iteratively calculating the output values of the connected neurons starting at the input neurons. The input value net^N of a neuron N is the sum of its predecessor output values a^{N_i} multiplied by the synapses weights:

$$net^N = \sum_i a^{N_i} w^{N_i N} \tag{4}$$

The output value is than determined by evaluating the so-called activation function for the given input net^N. The connected neurons set up the network, which therefore is able to produce the n-dimensional output for a given m-dimensional input data.

First, the ANN has to “learn” a certain mapping in a training process. Afterwards, the ANN is able to produce an approximated output pattern for a given input pattern. Learning methods can be grouped by *supervised learning*, *reinforcement learning* and *unsupervised learning*. Extensive explanations about the theory of ANN, their application and different models are given e.g. in Kinnebrock (1994) and Zell (1994).

Settings

For the ANN-bathymetric model, a multilayer perceptron is used, which is trained by the well-known backpropagation learning-rule (supervised learning). A multilayer perceptron is a feedforward network, where the neurons are arranged in layers. The first layer consisting of input neurons is called input layer. The output layer is set up by output neurons and is located at the last position. Between these layers, an arbitrary number of hidden layers are placed. Each layer is completely connected with the subsequent one. This topology is depicted in figure 2 and will be denoted by an n-…-m-topology, where each variable stands for the number of neurons of that layer[3].

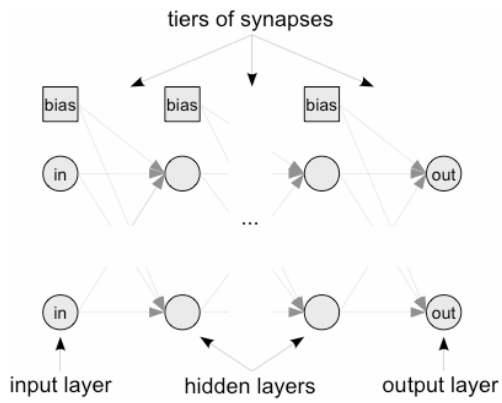

Fig. 2. Topological structure of a multilayer perceptron (MLP) consisting of an input, n hidden and an output layer. A MLP with n layers of neurons has n-1 tiers of synapses.

The logistic activation function is used to determine the activation value (output) for a given input value for each neuron. It is defined as

$$f_a^N(net) = \frac{1}{1+e^{-c \cdot net}} \tag{5}$$

where net is the input for Neuron N and c is a scalar parameter that influences the slope of the activation function. The shape of f_a^N is depicted in figure 3 for different parameters c.

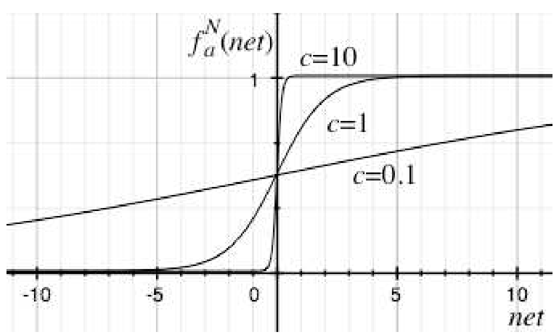

Fig. 3. Logistic activation function for $c = 0.1, 1, 10$. The neurons use this function with $c = 1$ to determine the activation value.

For $c \to \infty$, f_a is equal to the unit step function. For $c \to 0$, the logistic activation function approximates the constant function $f_a^N(net) = 1/2$. In this paper, all hidden and output neurons use the logistic function with parameter $c = 1$. Investigations for differing c have been done,

3 For example: 2-20-10-1-topology stands for a MLP with 2 input neurons, 20 neurons in the first hidden layer, 10 neurons in the second hidden layer and 1 output neuron

but it is still an open question if this parameter can be incorporated by an adequate learning rule.

The input neurons do not provide an activation function as such. The activation function serves as a transformation of the input values to the interval $[-1,1]$. To define this transformation function, the range of the input values has to be determined first. Hence, all input patterns have to be known a priori or the range has to be estimated at least. The following transformation function will be used for each input neuron:

$$f_t^N(v) = 2\frac{v - r_{min}^N}{r_{max}^N - r_{min}^N} - 1 \tag{6}$$

where $[r_{min}^N, r_{max}^N]$ is the range of input values v for neuron N.

The output neurons involve the logistic activation function plus an output transformation function. The transformation function for an output neuron maps the activation value of range $[0,1]$ to the given range of the output values $[r_{min}^N, r_{max}^N]$ of neuron N:

$$f_t^N(v) = v \cdot (r_{max}^N - r_{min}^N) + r_{min}^N \tag{7}$$

In the bathymetric context, input parameters are position and time, which are called x, y and t. The transformed values will be denoted as $\hat{x}$, $\hat{y}$ and $\hat{t}$. The following model studies in the third section are generally based on these transformed coordinates.

Empirical results

For evaluation of the ANN-bathymetric model two-dimensional training data of a domain around the Medem Channel near the German Bight was used as reference. The depth was measured at about 10000 positions in the year 1983. Figure 4 illustrates the linearly interpolated measured data on the basis of a triangulated mesh.

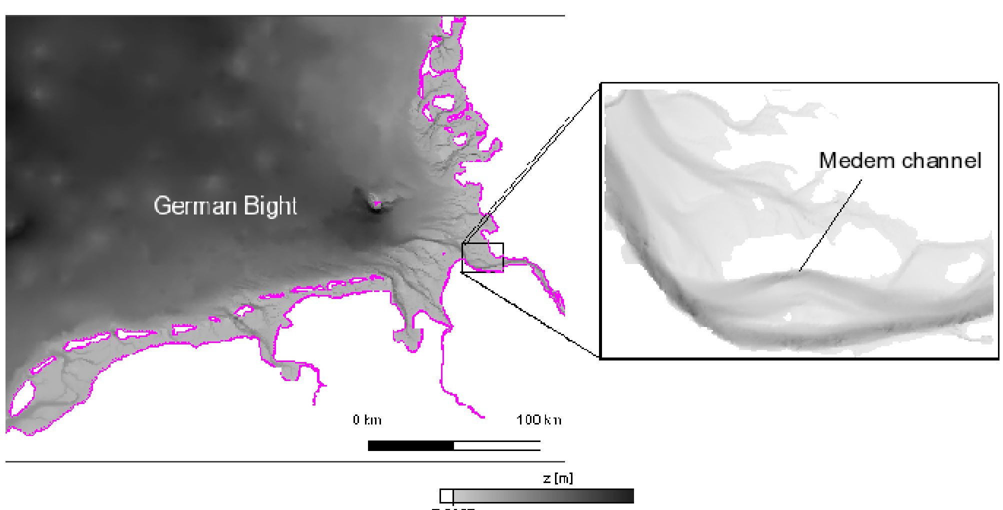

Fig. 4. Bathymetric data around the Medem Channel is used as training data for the ANN-bathymetric model. The data consists of about 10000 measuring points that were taken in 1983.

Since ANN is said to work like a "black box", the question on how to choose an adequate network topology for a certain set of training data is not easy to answer. In literature it is usually told that one has to figure it out by a trial and error process or by experience. Following this advice, many models with different network topologies were trained (most of the type 2-n-m-1). Figure 5 shows exemplary error plots.

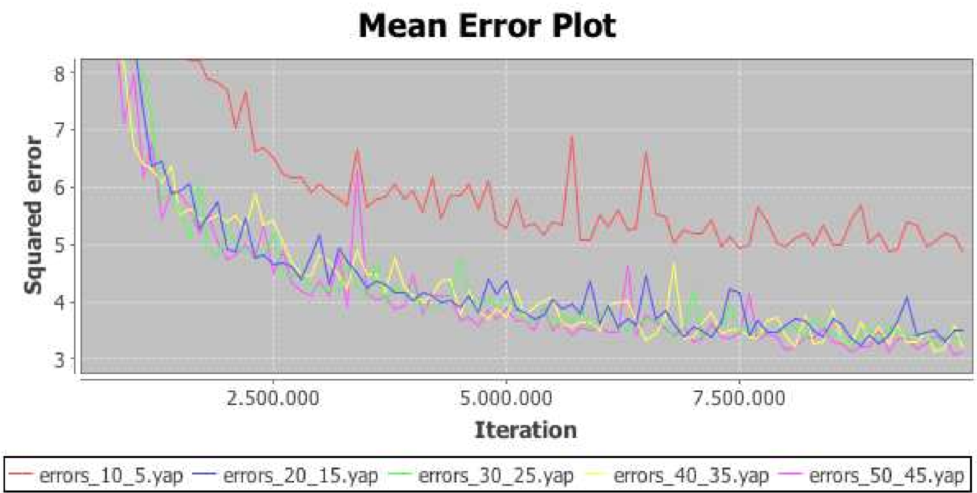

Fig. 5. Results of the empirical study for the regarded training data: the more neurons in the second and third layer, the smaller the training error.

These errors were averaged over five training processes for each topology. The weights of the synapses were reset randomly before starting the next training process. The error plots reveal that the training error decreases by increasing the number of neurons in the second and third layer. Bad convergence seems to be an indicator for an insufficient topological granularity.

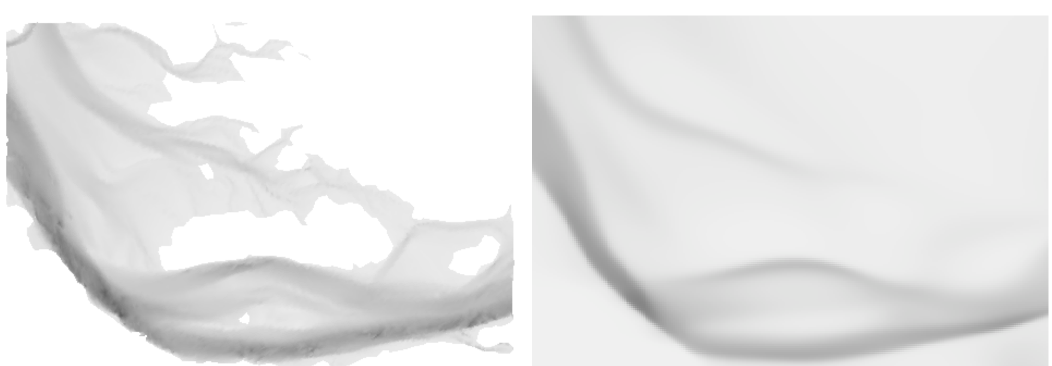

Fig. 6. Original bathymetry on the left and the approximated bathymetry on the right (resulting by an ANN using a 2-50-45-1 topology).

Figure 6 contrasts the original data (linearly interpolated on the basis of a triangulation) to the resulting bathymetry of an ANN using 2-50-45-1-topology. The coarse structures are approximated well and even in the domain of missing data a plausible "interpolation" is achieved. But the question is still open, how to choose an appropriate network topology in order to represent a mapping well.

LOOKING INTO THE BLACK BOX: THE RELATIONSHIP BETWEEN ANN AND BATHYMETRIC STRUCTURES

The previous section revealed that the network topology significantly affects the ability of an ANN to represent a mapping. This section deals with the problem, how an ANN maps or stores information on bathymetries and how this information can be retrieved from an ANN. Sufficiency of network topology is an important question concerning the representability[4] of an ANN.

Zell (1994) demonstrates the problem of linear separability. In his considerations the neurons use binary activation functions: a multilayer perceptron with two layers[5] is able to linearly separate the space of input parameters only. By introducing another layer, the network is able to represent a logical combination of linearly separated sets. Using four layers, i.e. three layers of changeable synapses, leads to a network that can represent sets with an arbitrary shape by logical combination of convex sets.

In the following, simplified bathymetries will be considered to demonstrate basic relations between bathymetric structures and network topology. The regarded bathymetries and bathymetric structures offer two values for the depth only, unless otherwise noted.

Basic Relations

Regarding the smallest possible configuration for an ANN in the bathymetric context, some basic relations between the ANN and its resulting bathymetry will be shown.

One neuron connected to its predecessor synapses acts as a border that linearly separates the input space as illustrated in figure 7. For an ANN-bathymetric model, the two dimensional input space consists of the x- and y-coordinate. Hence, such a neuron including its predecessor synapses can be interpreted as an *edge* of a bathymetric structure. The properties of the edge, which are the direction, position and slope, can be influenced as will be shown in the next three examples.

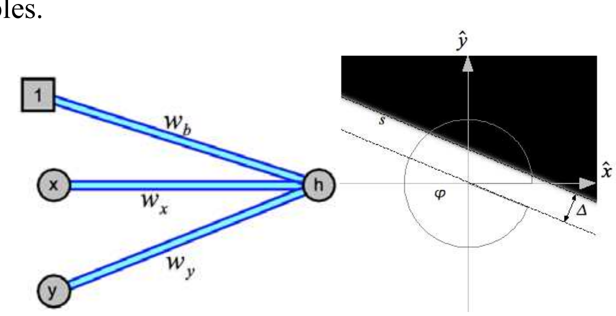

Fig. 7. A single neuron acts as an edge in the x,y-plane. The edge is characterized by the direction (expressed by φ), an offset (Δ) and the slope (s) depending on the weights w_x, w_y and w_b.

---

4 Representability of an ANN means if the ANN is able to represent a given mapping.
5 That means that the network consists of one layer of changeable weights of synapses.

Since the logistic function is used in this setting instead of the binary one, the resulting edge is not that sharp-cut but more diffuse. The slope is set by the parameter c of the logistic function as explained above. But the slope is influenced by the amount of the weights w_x and w_y, too. The factor s for the slope is given as

$$s = \sqrt{w_x^2 + w_y^2}\,. \qquad (8)$$

Figure 8 illustrates the influence of s: the higher the value, the sharper the edge.

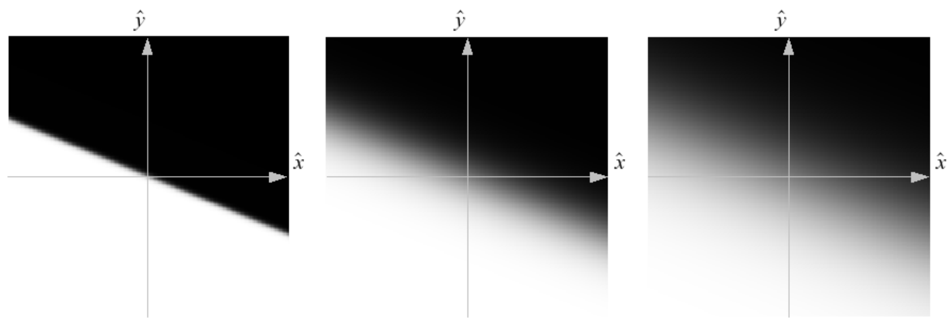

Fig. 8. The slope of an edge is influenced by the factor s. The parameters are $\varphi = 337.5°$, $\Delta = 0$ and $s = 100{,}10{,}5$.

The direction is another parameter of the edge. An angle φ can be defined for the direction against the ratio of w_x and w_y

$$\sin\varphi = -\frac{w_x}{\sqrt{w_x^2 + w_y^2}} = -\frac{w_x}{s}\,. \qquad (9)$$

In figure 9 three edges are shown for different angles.

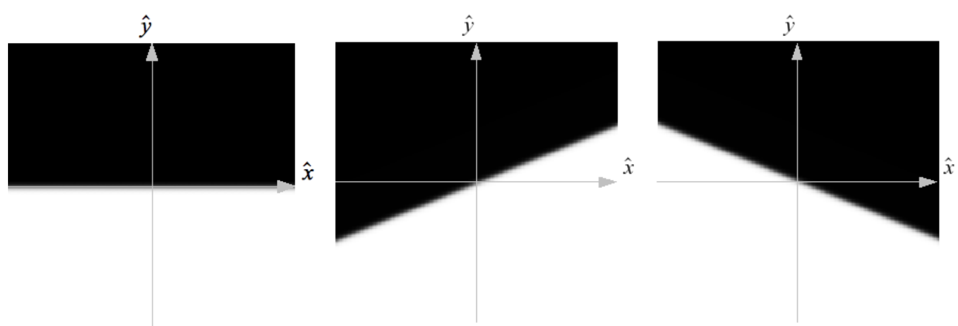

Fig. 9. The direction of an edge is influenced by the angle φ. The parameters are $\varphi = 0°, 22.5°, 337.5°$, $\Delta = 0$ and $s = 100$.

Due to the transformation function defined in (6) the origin of the transformed input values ($\hat{x}=0,\ \hat{y}=0$) is located in middle of the domain. Until now the edges all run through this origin. An offset can be achieved by introducing a so-called bias neuron that constantly returns the value one. The offset is then defined as the ratio of the amount of w_b and s

$$\Delta = \frac{w_b}{s} = \frac{w_b}{\sqrt{w_x^2 + w_y^2}} \tag{10}$$

where w_b is the weight of the synapse that connects the bias neuron with the regarded neuron. Figure 10 clarifies the influence of this parameter. For the given transformation function in (6), Δ should be in the interval $[-1,1]$, since other values would position the edge out of the given input ranges.

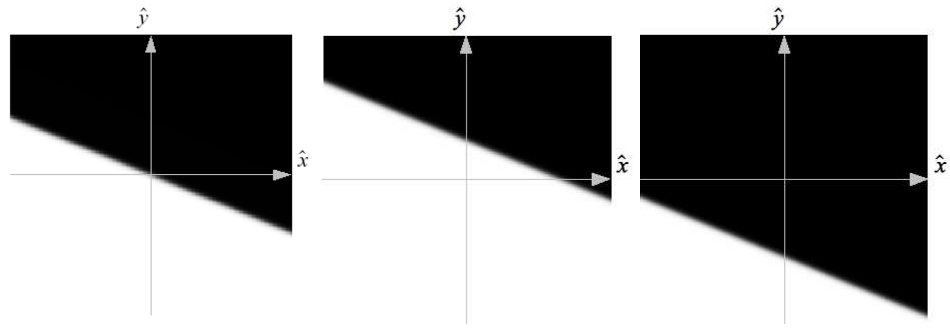

Fig. 10. The offset is set by the ratio Δ. The parameters are $\varphi = 337.5°$, $\Delta = 0,-0.25,0.5$ and $s = 100$.

Finally, the regarded network topology is sufficient to describe an ascent of variable slope, direction and position.

Representation of characteristic structures

Besides the ascent further characteristic geometric structures in bathymetries are for example the sea-channel, the islands or the tideland. Such structures can be modeled as a combination of multiple edges. Thus, multiple neurons are required in the first layer of the network topology. The combination of these edges is then mapped by the following layer and accordingly the second tier of synapses. The following examples illustrate a simplified sea-channel represented by two edges (figure 11) and an island set up by four edges (figure 12).

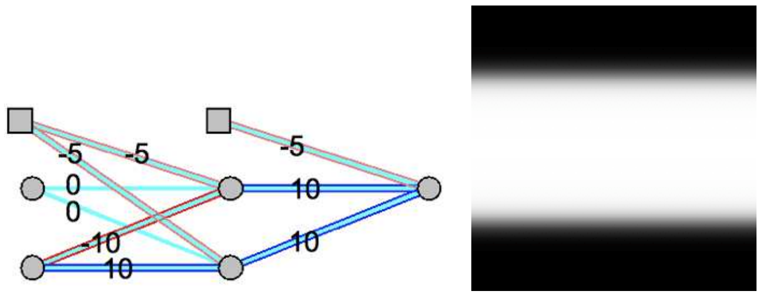

Fig. 11. Representation of a simplified sea-channel structure by the given network. Weights are plot onto the synpases.

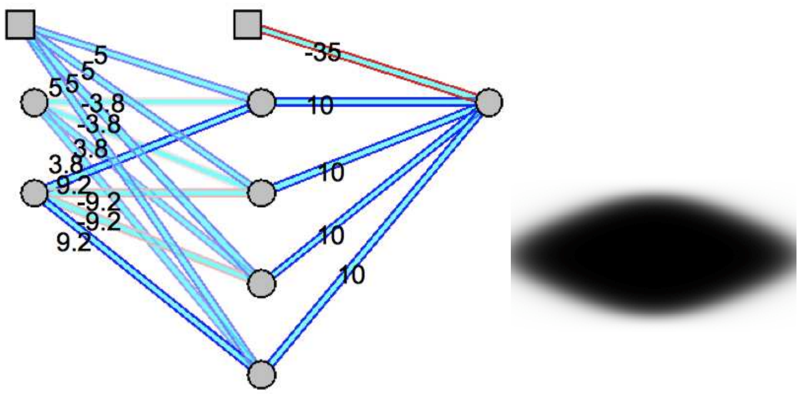

Fig. 12. Representation of a simplified structure of an island by the given network. Weights are plot onto the synpases.

Regarding the network topologies that produce the resulting bathymetric structures the following conclusion can be drawn.

First of all, every edge in the bathymetric structure is represented by one neuron in the second layer and the appropriate synapses in the first tier as described in section “Basic Relations”. For the topology of the sea-channel both edges are independent of parameter x (and $\hat{x}$ respectively). Thus, the edges run horizontally. Further the factor s for the slope is 10 each and both edges are offset to half of the way of the defined range (offset from the origin at $\hat{y} = 0$ to $\hat{y} = \pm 0.5 = \frac{w_b}{s} = \pm \frac{5}{10}$).

Secondly, these edges are now combined by the neuron[6] in the third layer. Looking at the weights of the synapses in the case of the sea-channel (figure 11), this combination corresponds to a simple logical connective, a disjunction. Figure 13 shows the disjunction of two “edges” in the 1D-case. The logical “or” of edge one and edge two has the value one as soon as the value for one single edge is one (approximately). In the case of the island

6 The edge is represented again by the neuron in interaction with the connected predecessor synapses (here located in the second tier).

structure the combination corresponds to a conjunction, that only results in the value one in the domain, where all activation functions of the predecessor neurons compute the value 1 (approximately) depending on x and y.

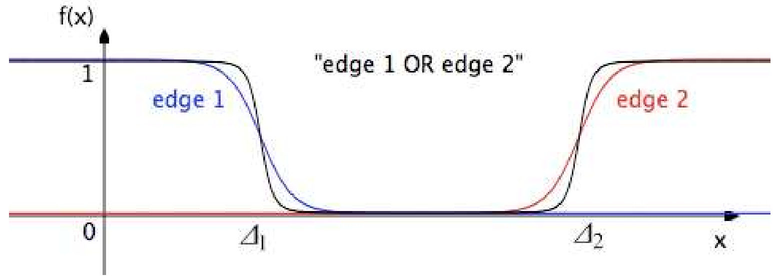

Fig. 13. The disjunction of two edges "edge 1" and "edge 2" in the 1D-case for simplification.

Identification
As seen in the previous section a bathymetric structure can be created by setting up a network with appropriate topology and weights. The other way round a network might be given, that "has learned" some bathymetry after a training process with appropriate data. Here, the question arises if it is possible to identify a neuron that "is responsible" for or represents a certain edge in the bathymetric structure. For this purpose preliminary investigations have been done, where only the second layer of neurons was taken into account up to now. For simple geometric structures the identification was successful, as shown in figure 14.

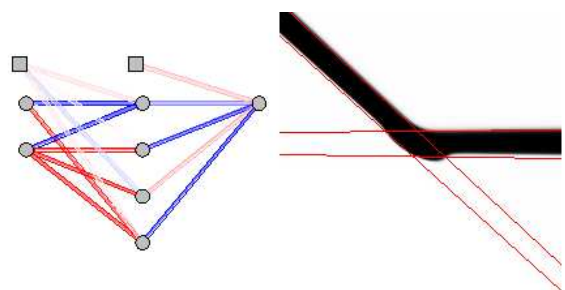

Fig. 14. Identification of the neuron, that represents a certain edge of the resulting bathymetry. Along the a red line a certain neuron of the second layer provides the value 0.5.

As seen in figure 15 more effort is required for more complex structures. The rest of the topology has to be taken into account to. At first, the edges are not "active" in the whole domain, but they are something like cut-off due to the combinations carried out in the third layer. Furthermore, some edges seem to be offset by a not yet determined action in the network. Therefore further investigations have to been done.

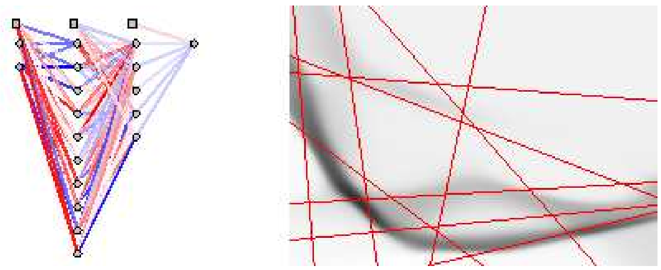

Fig. 15. Identification of neurons for a more complex structure. The edges (marked red) do not fit to the bathymetric structures.

Conclusions and Outlook

This paper is based on the idea of setting up a bathymetric model by using an artificial neuronal network. Good results could be achieved with a multilayer perceptron for the time independent (2D) and time dependent (3D) case (see section 2 and Berthold and Milbradt (2009) respectively). Especially the choice of network topology drastically influences the quality of the resulting bathymetry. This important choice of network topology is usually based on experience, because ANN is said to act like a "black box".

Here, the question arose how an ANN maps or stores information on bathymetries. By looking into this "black box" in section three basic relations between the geometric structures of time dependent bathymetries and the topological structure of ANN were regarded. It was demonstrated that a single neuron in the second layer and the weights of its predecessor synapses represents an *edge* in the resulting bathymetry. Considering this, simple bathymetric structures were modeled successfully. Besides topological structures for the 2D case, time dependent bathymetric structures have to be investigated.

Coming from a bathymetric structure, "responsible" topological structures in the second layer could be identified successfully. For more complex bathymetric and topologic structures the other layers have to be regarded as well to achieve a successful identification. More investigations have to be done here in future.

REFERENCES

Berthold, T. and Milbradt, P. 2009. Artificial Neuronal Networks in Environmental Engineering: Theory and Applications. *Proceedings of 18th International Conference on the Applications of Computer Science and Mathematics in Architecture and Civil Engineering.* Ed. K. Gürlebeck and C. Könke, ISSN 1611-4086.

Kinnebrock, W. 1994. *Neuronale Netze – Grundlagen, Anwendungen, Beispiele*. R. Oldenburg Verlag*,* Munich, Vienna.

Zell, A. 1994. *Simulation Neuronaler Netze*. Addison-Wesley.