European Water 39: 3-13, 2012.
© 2012 *E.W. Publications*

Radionuclides migration modelling using artificial neural networks and parallel computing

O.S. Hilko*, S.P. Kundas and I.A. Gishkeluk
Laboratory of Information Systems and Technologies in Ecology,
International Sakharov Environmental University, Minsk, Belarus
e-mails: o.s.hilko@mail.ru, kundas@iseu.by, gishkeluk@iseu.by

Abstract: In the paper the result of application of artificial neural networks (ANN) for radionuclides transport modelling with surface runoff is presented. ANN with supervised training based on back propagation algorithm was used to predict radionuclides transport in the soil and on its surface. Application of ANN for substances migration modelling is worth using, because it works like a “black box” and can catch patterns, which are hard to formalize in mathematical form or can’t even be introduced in the form of equations. Factors, which influence mass transfer process in different types of soil and on the soil surface, were defined for micro and meso/macro scales. They were introduced in a numeric form, normalized and used as input layer neurons of ANN. The obtained radionuclides concentration was used as neuron of an output layer. ANN is an example of empirical modelling approach and requires computing operations of the same type with a large number of field measured datasets. That is why the ANN training process took a lot of time. This problem was solved using parallel computing, which had a sufficient accelerating effect due to the nature of the training process. One of the technologies for parallel computing is CUDA from Nvidia, which uses a video card graphical processing unit. CUDA-technology was applied to accelerate ANN training and computing based on back propagation algorithm, which runs on a CUDA-device up to 5 times faster than the same operations on a CPU (central processing unit).

Key words: radionuclides migration modelling, soil surface, artificial neural networks, ANN, back propagation, NVIDIA CUDA

1. INTRODUCTION

Radionuclides transport modelling is quite a complex task. But it is very important nowadays due to the man-triggered existence and pollution. There are several approaches applied in this science area. Some of them are founded on the phenomenological theory of mass transfer and the laws of thermodynamics based on differential equations (the convective diffusion equation, St. Venant equations, diffusive and kinematic wave equations (Borah, 2003), balance equation of the slope runoff, etc.). Others are empirical models, which require large sets of experimental data in special conditions (Kundas, 2011).

Empirical models, describing the horizontal displacement of matter on the soil surface are based on natural observations. They include: the universal Ushmeyer-Smith’s equation of soil erosion (Wischmeier, 1978) (USLE); its MUSLE and RUSLE modification, developed by Larionov (for predicting migrating of radionuclides from storm water runoff); EPIC (Williams, 1989) and others.

Erosion process models and models of mass transfer with surface water, which are based on physical laws (Borah, 2003; Favis-Mortloc, 2007), include: WEPP, NonPoint Source pollution model (AGNPS), EUROSEM, STREAM, LISEM, Soil and Water Assessment Tool (SWAT), Dynamic Watershed Simulation Model (DWSM), KINematic runoff and EROSion model (KINEROS), CASC2D, models of Mirtskhulava, Kondratyev, Sukhanovskaya, Svetlichny, multi-purpose computer systems Areal Nonpoint Source Watershed Environment Response Simulation (ANSWERS), Chemicals, Runoff and Erosion from Agricultural Management Systems (CREAMS), the European Hydrological System model (MIKE SHE), Hydrological Simulation Program – Fortran (HSPF) and others.

Empirical models are mainly used to predict the environmental situation in a short period of time and make the analysis of separate factors' influence on the studied process complicated or even impossible.

The second modelling approach, which is based on physical laws, involves studying the governing processes and describing their dynamics with physically interpretable equations. As a rule, the systems based on differential equations, such as phenomenological models, represent the balance ratio of matter and energy in the system. This approach does not require a redundant set of experimental data to determine the parameters, which are used as variables. Most of the used parameters have clear physical or biological meaning, and therefore can be measured directly.

In the case of physically based approach determination of static coefficients, which are used in differential equations, are required. Computing of the coefficients is a very complicated task, because of the following reasons: 1) several physical values should be measured for a sufficient amount of time; 2) most of coefficients can be used in modelling of only those catchment areas they have been measured (or calculated) for.

Thus, most of the developed models greatly simplify the actual process and don't take into account the influence of separate factors; or they take the most influencing factors into consideration to decrease modelling error, what leads to the complication of the model and simulation, creates problems of numerical implementation and of the lack of input parameters in database.

An alternative tool for solving such problems is use of an artificial neural network. ANNs work like a "black box". They are usually used to model complex relationships between inputs and outputs, which are hard to formalize in mathematical form or can't even be introduced in the form of equations, or to find patterns in data.

There are several examples, where ANNs were used in mass transfer modelling and simulation of surface runoff distribution. Generalized Regression Neural Network (GRNN) model was used to define the influence of rainfall and surface drainage water on nutrient load into the neighbouring water systems by predicting the surface water quality and quantity (Kim, 2006). In Kanevski (1997) ANN and modern geostatistical models were used for the analysis of spatially distributed environmental data. Runoff and sediment yield from an Indian watershed during the monsoon period were forecasted for different time periods (daily and weekly) using the back propagation artificial neural network (BPANN) modelling technique (Agarwal, 2009). In de Vos (2005) multi-layer feed forward ANNs were used for rainfall-runoff modelling in the Geer catchment (Belgium).

ANNs have the following advantages (Haykin, 1999): the ability of generalizing and self-training, solving problems with unknown patterns, resistance to input data noise, adaptation to a changing environment, fault tolerance at the hardware implementation of neural networks, scalability, consistency of analysis and design, etc. ANNs have also some drawbacks (Haykin, 1999). Intermediate results can't be interpreted. A large training set is needed to construct an adequate model. The process of network training takes a lot of time. ANNs may have a highly complex internal structure. There is no rigorous theory on the ANN structure choice.

However, due to using parallel processing technology, which greatly accelerates the speed of network training, and in case of having a sufficient database for network training (a large quantity of experimental data on the concentration and activity of radionuclides is available; data on the state of climate, soil and relief have been accumulated) application of artificial neural networks in radionuclide transport modelling is appropriate. Choosing different topologies, combinations of activation functions and training parameters for hidden layers, several ANNs with minimum error can be selected and used for practical purposes.

In this paper, ANNs with supervised training based on back propagation algorithm were used to predict the vertical and horizontal transfer of radionuclides in the soil. This algorithm was chosen because it can be easily paralleled and it is not difficult to change the structure of ANN if new influencing factors are added (ANN doesn't need to be retrained from the beginning; it can be trained with new data on the base of the previous state).

Several ANN structures were developed to predict radionuclides migration on different scales.

ANNs were used to calculate result concentration (or activity) in a soil point or in a watercourse (alignment) at different time periods. That is why concentration was included as neuron in output vector. It is a common practice to use factors, which influence the process, as input neurons in ANN. So, it was needed to determine these factors and find dependences in governing processes for vertical and horizontal migration. The work was supported by the Belarusian Republican Foundation for Fundamental Research. In this paper horizontal radionuclide migration is particularly described.

2. ANN APPLICATION FOR RADIONUCLIDES MIGRATION IN HORIZONTAL DIRECTION

Contaminant migration in the horizontal direction is affected by the following processes: water erosion (flood runoff, snowmelt and storm water under the influence of gravitational forces), wind erosion (deflation), human activity, ion exchange, diffusion, capillary phenomena, sorption and desorption of soil particles.

Primarily radionuclide migration in an unsaturated zone and on the soil surface depends on physical and chemical characteristics of soil and radionuclide and the surface runoff distribution. The surface runoff and its distribution are affected by a wide range of factors (Kovalenko, 2006; Solomon, 2005; Panagopoulos, 2011; Paul, 1999; etc.):

- rainfall: intensity, duration, distribution, the amount and phase state,
- initial moisture content,
- the presence of vegetation, vegetation heterogeneity cover and its type,
- filtration and sorption characteristics of the soil (acidity, grading, cation exchange capacity, humus content, humidity),
- land development and the type of terrain,
- the level and steepness of the slope,
- relief type,
- wetlands,
- temperature,
- wind speed and direction.

Influencing factors should be divided into groups according to the scale of area the forecast should be made for. Traditionally micro (plots, slopes) and macro/meso (watershed) scale are differentiated.

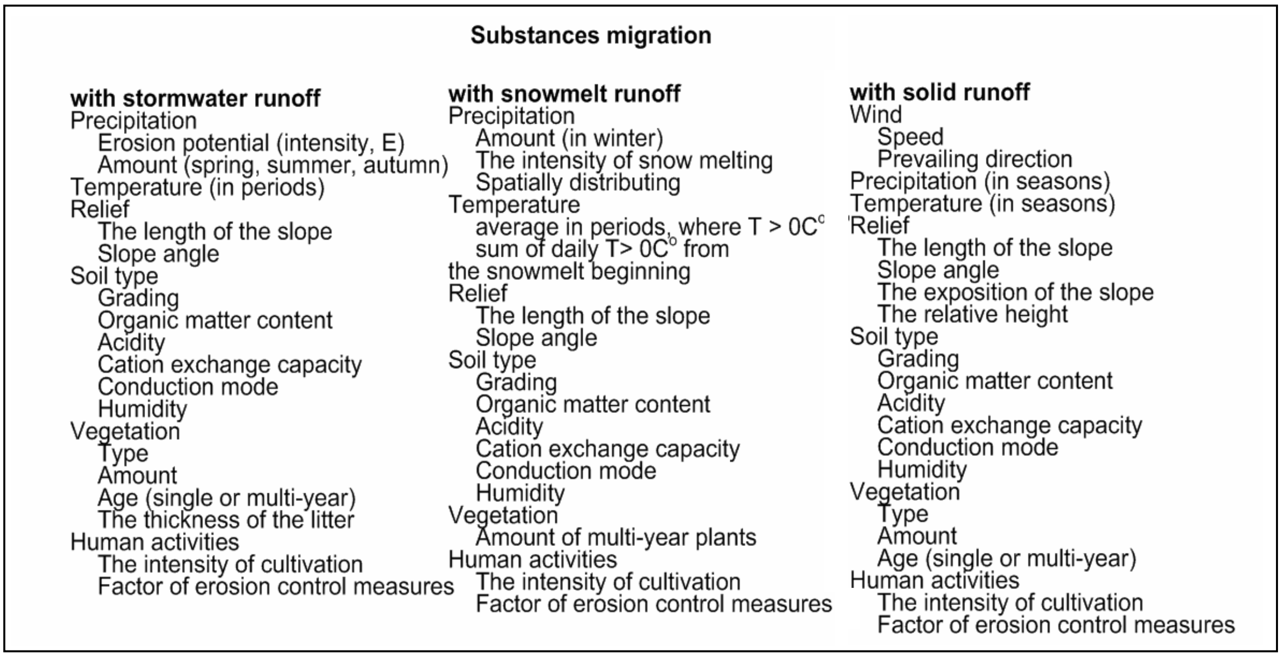
Micro scale modelling can be applied to predict secondary contamination of close located areas (redistribution of contamination including decay) and radionuclides redistribution between nearby spatial points. Macro scale modelling allows us to analyze migration of radionuclides from catchments to water objects (rivers of lakes).

2.1 Micro scale modelling

In micro scale all the factors affecting migration with surface runoff were carefully studied. Many authors (e.g. Blanco-Canqui, 2008; Shao, 2009; Kovalenko, 2006; Paul, 1999) have proposed to divide them into three groups, according to the erosion processes. These groups are listed below (see Figure 1).

The architecture of the project was developed in accordance with this approach. It is proposed to use three ANNs to predict radionuclides redistribution in micro scale. Each of the networks predicts the migration that occurs under the influence of some process: 1) wind erosion (with a solid runoff), 2-3) water erosion (with snowmelt and rainfall runoff). This approach gives the opportunity to obtain a continuous prediction of radionuclides concentrations on the soil surface at different distances from the site of substances' deposition.

In micro scale we took into account the influence of higher located neighbour points' radioactivity. ANN topology for radionuclide migration with rainfall runoff is chosen as example. Among the input factors (see Figure 1) there are time and spatial dependent ones.



Substances migration

with stormwater runoff
Precipitation
Erosion potential (intensity, E)
Amount (spring, summer, autumn)
Temperature (in periods)
Relief
The length of the slope
Slope angle
Soil type
Grading
Organic matter content
Acidity
Cation exchange capacity
Conduction mode
Humidity
Vegetation
Type
Amount
Age (single or multi-year)
The thickness of the litter
Human activities
The intensity of cultivation
Factor of erosion control measures

with snowmelt runoff
Precipitation
Amount (in winter)
The intensity of snow melting
Spatially distributing
Temperature
average in periods, where T > 0C^c
sum of daily T> 0C^o from
the snowmelt beginning
Relief
The length of the slope
Slope angle
Soil type
Grading
Organic matter content
Acidity
Cation exchange capacity
Conduction mode
Humidity
Vegetation
Amount of multi-year plants
Human activities
The intensity of cultivation
Factor of erosion control measures

with solid runoff
Wind
Speed
Prevailing direction
Precipitation (in seasons)
Temperature (in seasons)
Relief
The length of the slope
Slope angle
The exposition of the slope
The relative height
Soil type
Grading
Organic matter content
Acidity
Cation exchange capacity
Conduction mode
Humidity
Vegetation
Type
Amount
Age (single or multi-year)
Human activities
The intensity of cultivation
Factor of erosion control measures

Figure 1. Horizontal migration of radionuclides affected by different processes.

For single time step (week, month, year, etc.) one training data set for each pair of neighbour points (the distance *r* between them is closer than given distance *R*), which are on different height levels on the soil surface, includes the following inputs:

- start concentrations (Bq/kg) or densities (Bq/m^2) $C_{i\ start}$ and C_{start} at both points for the beginning of modelling period,
- sinuses of slope angles (sinα, sinβ) at both points (each is calculated by difference in height and distance) and distance (r) between them,
- total precipitation (in mm), average temperature (in degrees) and relative humidity (%) for time step (if the time step is greater than one year precipitation, temperature and humidity are required to be given by seasons),
- conduction mode factor, factor of human activity and factor of erosion control measures (as coefficients),
- spatially dependent characteristics for both points in pair [vegetation characteristics: vegetation type (as coefficient), radionuclide content in plants (Bq/kg), thickness of the litter (sm); and soil characteristics: percentage of sand, silt, clay; humus content (%), acidity (pH), cation exchange capacity (meq+/100g or cmol+/kg), average humidity (%) (see Figure 1)],
- constant coefficient of unaccounted factors' influence (0.5 units).

Unaccounted factors influence (as constant coefficient) was proposed to be included in ANN topology to take into account the possible impact of any additional factors. After ANN is trained, weight coefficients, which show the influence of input neurons on the first hidden layer, are calculated. Due to their values it is possible to conclude, how pronounced the impact is. Thus, analyzing the weight coefficients, which are associated with the last significant neuron in input layer (this neuron shows the influence of unaccounted factors), it is possible to conclude how well the ANN topology covers governing process.

In the micro scale while the analysis of spatial data is held, several pairs of points with one basic (pending) point can be selected (N_i – number of pairs). For one basic point only a single pair can be chosen to be included in the training set. The pair is selected by the greatest influence on the pending point. The selection is made by the maximum value of criteria, which is calculated for each *i*–pair:

$$\frac{C_{i\ start}}{C_{start}} sina_i \frac{R-r_i}{R} \quad , \ i=\{1,\ ...,\ N_i\} \tag{1}$$

where $C_{i\ start}$ and C_{start} – concentrations at both points in each pair for the beginning of modelling period (C_{start} – in pending point, $C_{i\ start}$ – in other point *i*–pair), *sin*α_i – sinus of surface angle, calculated by the distance between points in *i*–pair (r_i) and their difference in height (Δh_i):

$$sin\alpha_i = \frac{\Delta h_i}{\sqrt{r_i^2 + \Delta h_i^2}} \ . \tag{2}$$

In training data set final concentrations (Bq/kg) or densities (Bq/m^2) for the modelling period (time step) in each point are used as output neurons.

In computing mode for the case of modelling for several time steps, final concentration (or density) for the previous period is used as input (start) concentration (or density) for the next period.

2.2 Watershed scale modelling

Macro scale factors are influenced by micro scale ones, but differ in some way (some micro factors can't be measured in macro scale, some macro factors have no meaning in micro level) or have another time or spatial numerical measurement.

In macro scale it is quite hard to divide factors into groups in accordance with influence processes. Factors, which affect radionuclides on macro level from catchment to watercourse, are the following:

a) watercourse characteristics: concentration in input flow (Bq/l) (if watercourse starts out of the pending catchment), concentration in sediments (Bq/kg), hydrographical length of the watercourse (km), average slope of the watercourse (as sinus function of angle), height of the catchment above sea level (m), conduction mode (as moisture coefficient);

b) left and right watercourse bank characteristics of catchment: initial average pollution density (kBq/km^2); average slope of catchment hills (as sinus-function of angle); average relative percentages (%): of forest catchment, water logging, lakes and artificial ponds, stagnant lakes, karst catchment, plowing catchment to the total catchment area; characteristic of the soil type in catchment (texture (%): clay, silt, sand and rock; acidity (pH), average organic matter content (%)); average depth of the groundwater (the first of the aquifer); characteristic of the terrain (flat, rock);

c) climate characteristics: total rainfall (mm), average temperature (in degrees), average relative humidity (%), shift perpendicular of the watercourse to the two predominant wind directions (as cos(β–γ), where β – wind shift relative to the western direction, γ – shift perpendicular of the watercourse to its latitude) and their average speeds (m/s).

In macro scale one ANN is used. All mentioned factors are used as input neurons. The watercourse is divided to parts between targets with known concentrations in water and sediments. Each part has one training dataset for a single time step.

Pending radionuclide concentrations in the water (Bq/l) and in sediments (Bq/kg) for the end of modelling period are used as output neurons.

3. CUDA-TECHNOLOGY APPLICATION FOR SUPERVISED TRAINING OF ARTIFICIAL NEURAL NETWORKS

While ANNs were trained it was found, that training process took a lot of time. This problem was solved using parallel computing. The feasibility of parallel computing with a GPU (Graphics Processing Unit) using CUDA-technology was investigated. CUDA (Compute Unified Device Architecture) was used to accelerate ANN training based on back propagation algorithm. CUDA application in forecasting mode also made this process faster (Hilko, 2010b).

The mechanism of parallel ANN training algorithm based on back propagation includes a single-step calculation for all segments of the network. So if output parameters of the previous layer (for a prediction mode) or the next layer (for training) are unknown, it is impossible to calculate parameters of the current layer. There are two ways how the algorithm can be parallelized (Hilko, 2011): 1) simultaneous training of several ANN and choosing of the best one according to the minimum error criteria, 2) simultaneous calculation of several neuron parameters of one layer. The second method has been chosen to implement parallel training algorithm. Since it requires less memory on the device than the first one, this approach can be used for ANN, where layers have larger number of neurons.

The training algorithm consists of three blocks, repeated for each set of input data in the following order: 1) direct computing (the calculation of each layer outputs, starting with the first), 2) calculation of error for the whole network (root mean square error or mean relative error) and errors of each layer, starting with the last one, 3) calculation of correction values for the weights' matrices and thresholds' vectors with their further updating.

Because of the hardware features of video cards the number of neurons in the layer must be a multiple of 2^n, where $n = \{1, \ldots, 5\}$. For those networks, where the number of neurons in layers is greater than 8, the number of neurons must be divisible by 16 (Hilko, 2010a). For this purpose weights' matrices (W_i) and their correction values (ΔW_i), the input (Y_{i-1}) and output (Y_i) vectors, thresholds (T_i) and their correction values (ΔT_i), the current error vectors (E_i) are aligned (filled with zeros) with the selected dimensions ($i = 1 \ldots N$, where N is the amount of hidden layers plus the output layer). When parameters are aligned "imaginary" (0-value) neurons are added to the ANN to get the amount of neurons divisible by 16. Neurons of *Inputs* and *Outputs* should be normalized in the range of (0;1).

It is important to note, that this approach should be mathematically checked for each type of the activation function. It is true, when one of the following conditions is met:

- activation function for 0-point is F(0)=0 (such as bipolar sigmoid function or hyperbolic tangent) and desired values of the *Outputs* "imaginary" neurons (Y_i', where $i = N$) are filled with 0;
- the desired values of the *Outputs* "imaginary" neurons are equal to F(0), so $Outputs_{im} = Y_N'{}_{\mathrm{im}} = F(0)$.

The scheme of interaction between the external and the developed (as a dynamic linked library) modules is shown in Figure 2 (Hilko, 2011). Baseline data on the current network state (W_i, ΔW_i, T_i, ΔT_i, E_i, Y_i for each layer) are formed by an external module according to the rules mentioned before.

If the network is trained for the first time, W_i and T_i are filled with random numbers in the range of $[-0.05;0) \cup (0;0.05]$ or other mathematical algorithms (e.g. genetic) are used. Before training starts, corrective weights and thresholds (ΔW_i and ΔT_i), the current error vector (E_i) and output vector (Y_i) are filled with zeros.

If the network has been trained before and needs training once again starting with its previous state, its saved values W_i, ΔW_i, T_i, ΔT_i, Y_i are used.

The current network state parameters are sent by reference to the control module (host). Some memory on the device is allocated for them. Memory is also allocated for the arrays, which are needed for intermediate calculations. At the first step the training datasets (*Inputs*, *Outputs*) included in one epoch are transmitted to the plug-in manager, at the second step – are put to the device.

Computing algorithm includes the consistent implementation of the direct and reverse passes for each inputs and outputs (*Inputs*, *Outputs*) set of one epoch. The control module copies the value of the output layer's root mean square error (RMSE) from the device, which is used in calculating of the total RMSE of each epoch. It is saved in a file. According to the epoch RMSE it can be judged, how well has the network been trained. When the training is completed (a required level was achieved or a specified number of iterations was done), the device returns current network state parameters W_i, ΔW_i, T_i, ΔT_i, E_i, Y_i for each layer back to the control module. Then these data are sent to the external module.

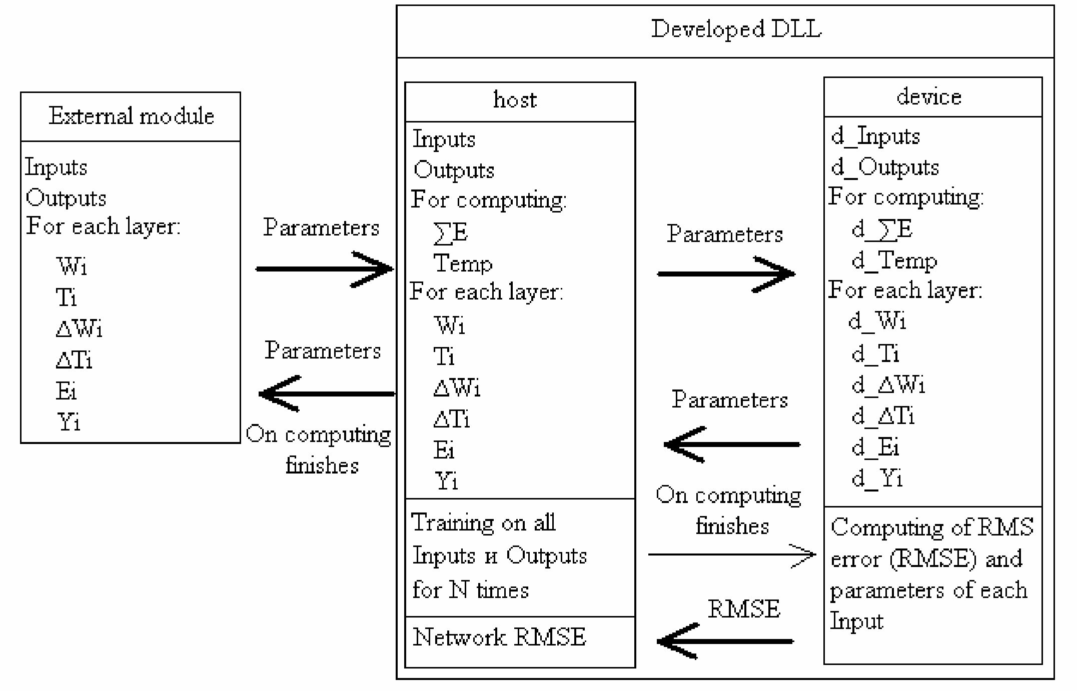


Figure 2. Scheme of interaction between external and developed modules.

Thus, there is no need for constant training parameters moving between the host and the device. Having once received them, the device trains the network itself, periodically returning only the value of RMSE to allow the current ANN state tracking. Also W_i and T_i for each layer can be transferred to use them in the external graphics module that visualizes current ANN state (see Figure 3), if it is necessary. However, it increases the training time. Graphic display of network status should be used to determine its topology, when parameters' visualization allows changing the structure to minimize the RMSE.

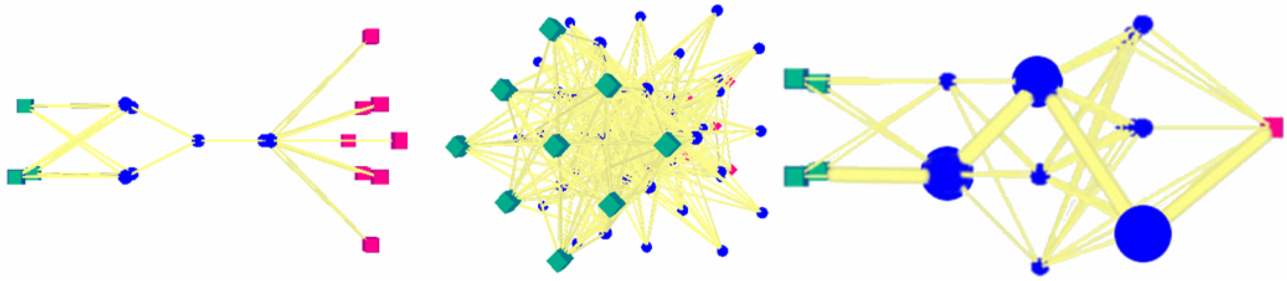


Figure 3. Examples of the current ANN states.

4. FEATURES OF MEMORY ALLOCATION

While using CUDA for ANN training is was found that ANNs with large amount of neurons can't be located on video card because of the lack of memory. For this reason we had to investigate the algorithms of memory allocation and to find out dependences between memory volume of the device and ANN topologies, which can be located on video card.

It is necessary to take into consideration that the video card memory for the ANN parameters is not completely allocated for them. It was experimentally established that the video card with 256 Mb full amount memory allows to allocate only 230 Mb (242 077 696 bytes – 90.2 %), the rest of memory is reserved. A 512 Mb memory video card allocates 473 Mb (92.5 %). In the 1024 Mb memory video only 951 Mb (92.9 %) are available. Moreover, during calculation of small arrays (eg. 256 elements) the memory is allocated from the reserve (Hilko, 2010b).

On the GeForce 9500 GT the function, which allocates memory for 3 vectors (dimension *m*) and 3 of the matrices (dimension *n*, *m* x *m* = *n*), was investigated in detail. Global memory for vectors starts to be allocated for $m \geq 1024$.The first vector memory is allocated with the required volume. The second one and others take a multiple of 64 Kb (65536 bytes) volume of memory.

For matrices up to (1024 x 1024) the situation is similar. However, more memory is allocated in the first matrix of a large size (1024 x 1024 and above), than it is required (see Table 1). Equation (3) was empirically obtained on experimental data for the matrices in Table 1:

$$Memo = (2\sqrt{n} - 1)^2 + 2^{16} - 1, \tag{3}$$

where *Memo* – is the volume of allocated memory in bytes for the matrix, *n* – is number of elements in the matrix. For other matrices (except the first one) memory is allocated as much as needed.

Table 1. Memory allocation for neurons weight matrices

#	Matrix Dimension	Number of elements, n	Required memory, bytes (col.2 x 4)	Allocated memory by CUDA, bytes	Memory by *(3)*, bytes	Attitude col.3/col.4
1	1024 x 1024	1048576	4255744	4194304	4194304	1,01468
2	2048 x 2048	4194304	16834560	16777216	16777216	1,00341
3	3072 x 3072	9437184	37801984	37748736	37748736	1,00141
4	4096 x 4096	16777216	67158016	67108864	67108864	1,00073
5	5120 x 5120	26214400	104902656	104857600	104857600	1,00042

The quantity of training datasets that can be put into the device's memory for network training was calculated. Datasets with the number of neurons in input vector up to 2608 were investigated. Also formulas for the described approach were derived to estimate the maximum ANN topology satisfying the following conditions, which can be trained on the device depending on memory size:

a) the number of hidden layers, including the output layer, is equal to two (Haykin, 1999): one latent layer and the output layer;
b) the number of neurons in the first hidden layer is equal to 2N0 + 1, where N0 – is the number of input neurons, which follows Kolmogorov's theorem (Kolmogorov, 1957). Input data must be normalized and aligned (Hilko, 2011) to N0' before the beginning of training. That is why input neurons number should be N0' = 16n (n – natural number). For the first case the number of neurons in first hidden layer (N1') should be equal to N1' = 2N0' (as N0 < N0' for all N0, which are not multiplies of 16, and which implies 2N0 +1 < 2N0'). For the case, where N0 is a multiple of 16, N1' = 2N0' + 16;
c) the number of output neurons is set arbitrarily N2'= kN0 (k > 0, k – coefficient of proportionality) and should be also aligned.

As a result, we obtained four equations (eq. (4) – (7), where $N_0' = N$) to assess the maximum number of neurons in the two-layer aligned ANN ($N - 2N - N$: N neurons in input layer –$2N$ neurons in first latent layer – N neurons in output layer), which may be trained on different cards depending on the RAM volume. If *Epoch* – is the number of input data sets for one training period, *Memo* – is the required memory volume in bytes for putting all the data elements, then for N_0, which are not multiplies of 16, equation (4) for $k \leq 2$ and equation (5) for $k \geq 2$ are true. For those N_0, which are multiples of 16, equation (6) is for $k \leq 2$, and equation (7) is for $k \geq 2$:

$$16(k+1)N^2 + (40 + 16.25k + 4Epoh(k+1))N + 256 - Memo = 0, \tag{4}$$

$$16(k+1)N^2 + (32 + 20.25k + 4Epoh(k+1))N - Memo = 0, \tag{5}$$

$$16(k+1)N^2 + (160 + 148.25k + 4Epoh(k+1))N + 256 - Memo = 0, \tag{6}$$

$$16(k+1)N^2 + (168 + 144.25k + 4Epoh(k+1))N + 256 - Memo = 0. \tag{7}$$

For example, solving quadratic equation (4) for N (*Epoch* consists of 1000 data sets) with equal number of neurons in input and output vectors ($k = 1$), the device with 230 Mb of available memory (242 077 696 bytes) can be used for training the ANN (2624 – 5248 – 2624), since $N = 2627.5$.

But due to the device's memory allocation peculiarities, in practice, only (2608 – 5216 – 2608) network can be put in the memory (2608 neurons in the input and output layers, 5216 – in hidden layer). It should be noted, that locating all specified data on the device requires a little bit more memory than it is necessary.

Figure 4 shows graphs of the allocated memory (in Mb) on the devices (9500 GT, 9800 GT and 220 M) and the theoretically calculated memory requirements (CPU) by (4) to put all the data elements of two-layer aligned ANN ($N - 2N - N$) with different numbers of neurons in the layers. The "step" memory allocation on the GPU is clearly seen.

5. RESULTS AND DISCUSSION

CUDA application in ANN training algorithm for radionuclides migration modelling was tested. Tests were performed on a computer equipped with 4-core Intel Core 2 Quadra (2.86 GHz per core) and GeForce 9500 GT, and for GeForce 9800 GT and GeForce 220M GT. Figure 5 shows graphs of elapsed time in seconds for 10 training iterations of aligned ANN with two hidden layers ($N - 2N - N$) with 1000 data sets per iteration.

Mathematical algorithm calculating on a CUDA-device runs up to 5 times faster than the same operations on a CPU (central processing unit). Maximum performance benefit depends on the configuration, kernel frequency, video memory capacity, bandwidth, version of the installed driver.

6. CONCLUSIONS

ANN was applied for horizontal radionuclides migration modelling. This approach is worth using because of several reasons. Factors in numeric and symbolic forms can be taken into account. Their separate contribution can be counted. No physically based equations should be used, because ANN can catch patterns which are hard to formalize to mathematical form. ANN can be used for other areas or catchments, if it's additionally trained on new data.

Factors, which influence horizontal radionuclides migration in macro/meso and micro scales were determined. They are used as input neurons in ANN structures.

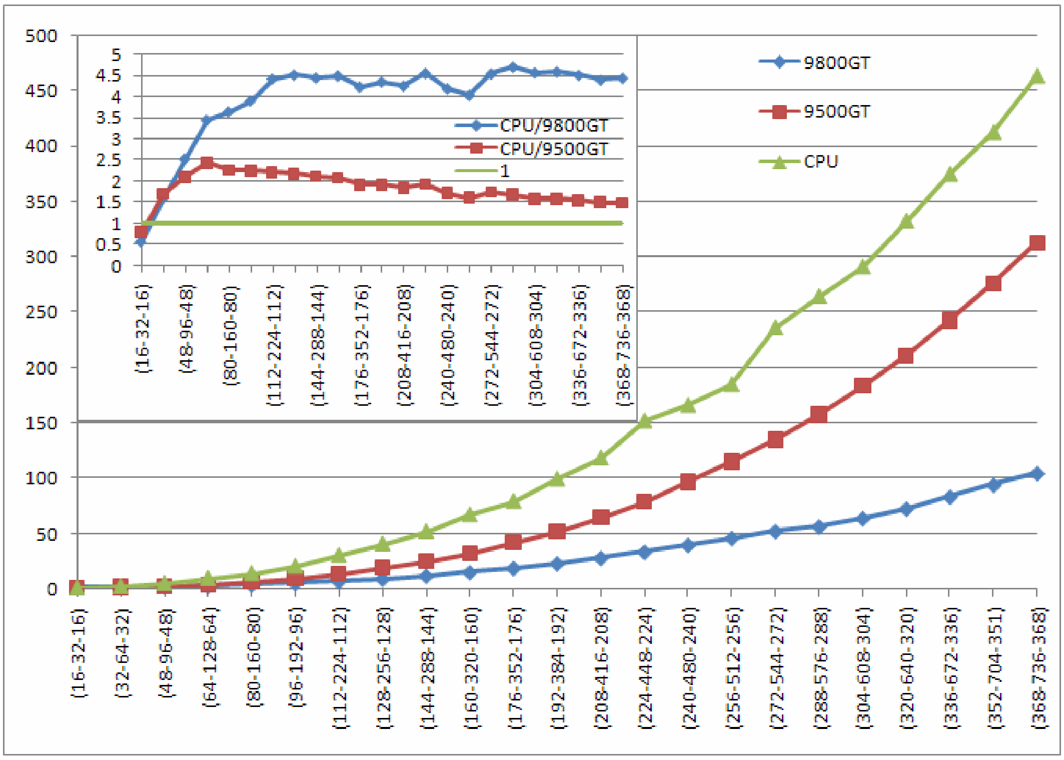


Figure 4. Memory allocation for different topologies of ANN.

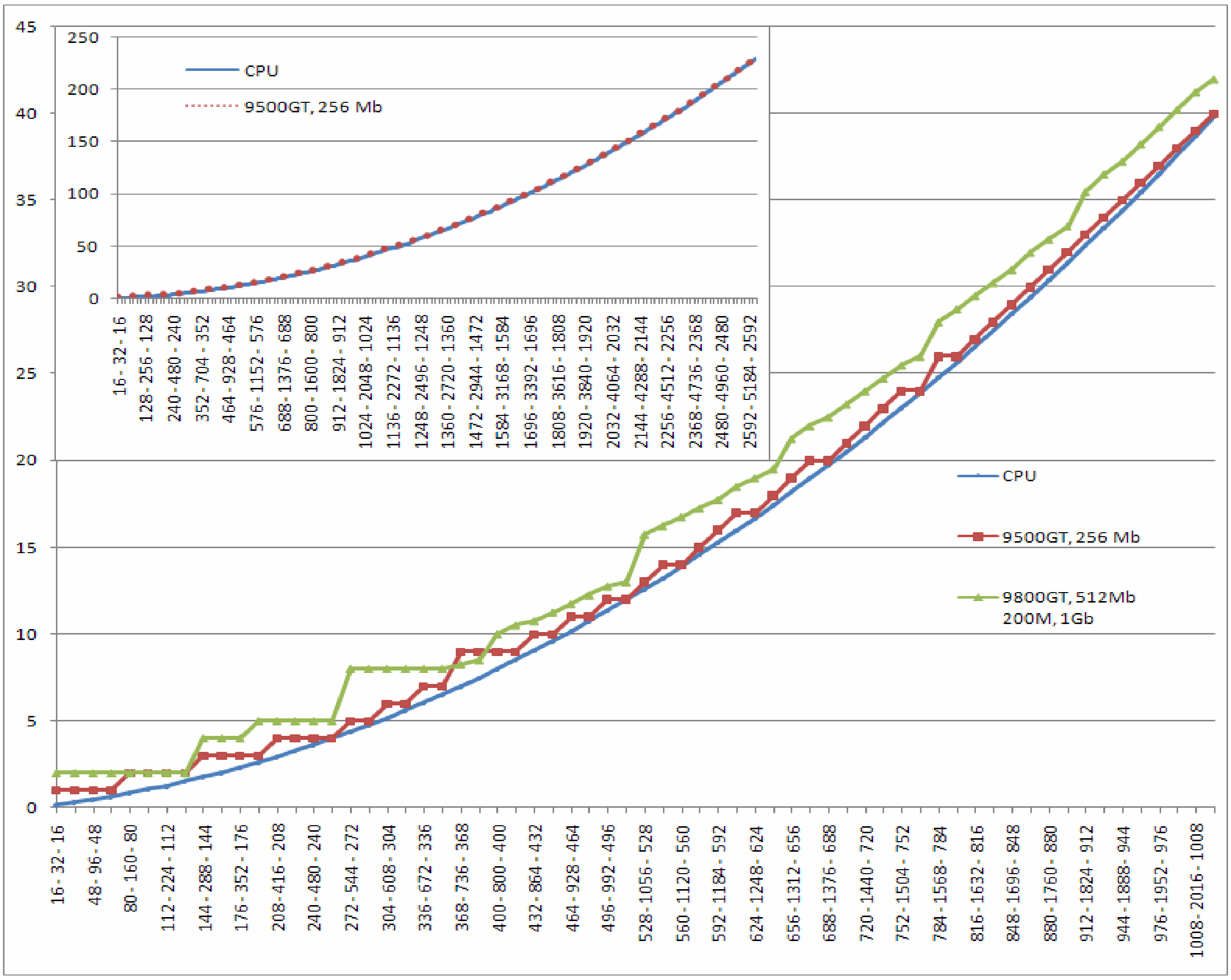


Figure 5. Elapsed time for two-layer ANN training.

Due to large amount of training datasets, ANN training process took a lot of time. This drawback was overcome by using parallel computing with CUDA from Nvidia. This technology is based on GPU computing and accelerates ANN training with a large number of neurons in layers up to 5 times. But due to the device's memory allocation peculiarities, limitations on RAM volume and necessity of parameters' alignment the data should be carefully prepared and analyzed before using.

It is possible to conclude, that CUDA-technology and ANNs are worth using not only in radionuclide migration modelling, but they can be used in a wide sector of ecological tasks, which require parallel computing.

REFERENCES

Agarwal, A., Rai, R.K., Upadhyay, A., 2009. Forecasting of runoff and sediment yield using artificial neural networks. Journal of Water Resource and Protection; 1: 368-375.

Blanco-Canqui, H., Lal, R., 2008. Water Erosion in *Principles of soil conservation and management*. Springer: 21-53.

Borah, D.K., Bera, M., 2003. Watershed scale hydrologic and nonpoint source pollution models: review of applications. Transactions of the ASAE; 46(6): 1553–1566.

Favis-Mortloc, D., 2007. Soil erosion models. http://soilerosion.net/doc/models_menu.html (accesses March 15, 2010).

Haykin, S., 1999. *Neural Networks: A Comprehensive Foundation*, Second Edition. Pearson Education, Inc.

Hilko, O.S., Kovalenko, V.A., 2010a. CUDA-technology application in artificial neural networks for solving ecological tasks (in Russian). Proc. of 10th Int. Conf.: Sakharov Readings 2010: Environ. Problems of the XXI Century. Minsk, Belarus, 2010: 87.

Hilko, O.S., 2011. CUDA-tecnology application for artificial neural network training (in Russian). Doklady BGUIR; 4(58): 55-62.

Hilko, O.S., Kovalenko, V.A., Kundas, S.P., 2010b. CUDA-technology application for parallel computing in artificial neural networks (in Russian). Doklady BGUIR; 7 (53): 83-88.

Kolmogorov, A.N., 1957. On the representation of continuous functions of several variables by superposition of continuous functions of one variable and addition (in Russian). Dokl. Akad. Nauk SSSR; 5 (114): 953-956.

Kanevski, M., Demyanov, V., Maignan M., 1997. Mapping of soil contamination by using artificial neural networks and multivariate geostatistics. Proc. of 7th Int. Conf.: Artificial neural networks - ICANN'97. Lecture Notes in Computer Science, Lausanne, Switzerland; 1327/1997: 1125-1130.

Kim, M.Y., Jee, H.K., Lee, S.T., Kim, M.K., 2006. Prediction of nitrogen and phosphorus transport in surface runoff from agricultural watersheds. KSCE Journal of Civil Engineering; 10(1): 53-58.

Kovalenko, V.V., 2006. *Modelling of hydrological processes* (In Russian). St.-Pb.: RGGMU.

Kundas, S.P., Gishkeluk, I.A., Kovalenko, V.A., Hilko, O.S., 2011. *Computer modelling of contaminant migration in natural disperse media* (in Russian). Minsk, ISEU.

Panagopoulos, Y., Makropoulos, C., Mimikou, M., 2011. Diffuse Surface Water Pollution: Driving Factors for Different Geoclimatic Regions. Water Res. Management.; 25(14): 3635-3660.

Paul, G.S., Hergarten, St., Neugebauer, H.J., 1999. Numerical simulation of surface runoff and infiltration of water. Process Modelling and Landform Evolution, Springer; 78: 93-107.

Shao, Y., 2009. Integrated Wind-Erosion Modelling in *Physics and modelling of wind erosion*. Springer; 37: 303-360.

Solomon, H., 2005. *GIS-Based Surface Runoff Modeling and Analysis of Contributing Factors*. A Case Study of the Nam Chun Watershed, Thailand. Thailand: Enschede, ITC.

de Vos, N.J., Rientjes, T.H.M., 2005. Constraints of artificial neural networks for rainfall-runoff modelling: trade-offs in hydrological state representation and model evaluation. Hydrology and Earth System Sciences; 2: 365–415.

Williams, J.C., 1989. The EPIC crop growth model. Trans. ASAE; 32(2): 497-511.

Wischmeier, W., Smit, D.P., 1978. *Predicting rainfall erosion losses – a guide to conservation planning*. USDA Handbook № 237, Washington.