

**Ya. I. Sokolovskyy, V. M. Shymanskyi, O. V. Mokrytska, Ya. V. Kharko***Ukrainian National Forestry University, Lviv, Ukraine*

NEURAL NETWORK MODEL FOR IDENTIFICATION OF MATERIAL CREEP CURVES USING CUDA TECHNOLOGIES

This paper addresses the problem of identifying rheological parameters of wood using artificial neural networks with parallel learning algorithm using Python programming language, Chainer framework and CUDA technology. An intelligent system for identification of rheological parameters of wood has been developed. The system created contains the most user-friendly interface, all the necessary set of tools for automation of the process of visualization and analysis of data. In the process of creation of the intellectual system, the following tasks were envisaged: to carry out the analysis of artificial intelligence systems and the analysis of training of artificial neural networks, in particular multilayer neural networks of direct propagation, recurrent neural networks and the Kohonen neural network; examine the structure of the Chainer framework and its interaction with CUDA; to conduct existing cloud technologies to accomplish the task; to conduct the analysis of algorithms of studies of artificial neuron networks, their mathematical providing; to implement parallelization of learning algorithms and to develop the necessary software. Using Chainer allows you to create a memory pool for GPU memory allocation. To avoid memory allocation and erasure during computing, Chainer provides the ability to use the CuPy memory pool as a standard memory allocation without dealing with memory allocation. An intellectual system to determine the physical and mechanical parameters of a mathematical model of non-isothermal moisture transfer and viscoelastic deformation of capillary-porous materials was developed. It provides the opportunity to identify parameters of the kernels of creep and relaxation that is written as a linear combination of exponential operators. The proposed algorithm of approximation and obtained calculated ratios of rheological behavior of wood by means of multilayer neural network with exponential activation functions in hidden layers allows to increase the accuracy of approximation of experimental creep data. The developed mathematical models can be used to create an automated systems of finite-difference calculation of temperature and moisture content, stress components during the drying of capillary-porous materials with taking into account the technological parameters of the drying agent.

Keywords: identification of parameters; parallelization; artificial neural network; python; Chainer framework; CUDA.

Introduction

Today, artificial neural networks have become widespread to solve a large class of tasks: information processing, first of all for identification, emulation, intellectual control, and predicting time series of arbitrary nature in the context of structural and parametric uncertainty. Particularly urgent is the problem of identifying the parameters of wood creep, which is the starting point for establishing the corresponding parameters of mathematical models of non-isothermal moisture transfer and viscoelastic deformation of capillary-porous materials during drying. The phenomenon of creep occurs in materials of different nature: wood, metal, plastics, concrete and others. The physical mechanisms of creep are individual for each of the materials. Creep studies also show that the process of this process is similar at first glance, and different materials require different approaches to describe certain phenomena that are characteristic of the material.

Neural networks represent promising computational technologies that provide new postulates to the study of control and analytical tasks. The use of neural networks allows to consider the factors based on which you can build short-term forecasts. Applying neural network architecture and the knowledge base, it is possible to obtain effective positive identification system parameters. To take into account external parameters, you must enable the corresponding input to the neural network. This uses the algorithm for determining the importance and significance of input

variables, with the exception of parameters that have little influence. On the rheological characteristics of wood is influenced by such factors as: temperature, humidity, size, breed and others. Definition of levels of impact factors on the behavior of the material is an actual problem that requires the involvement of advanced technology to solve, such as the use of artificial neural networks.

The construction of neural network model is adaptive without the participation of the expert in the learning process that allows you to create a system of decision-making. In particular, based on the numeric values of humid fields, the decision on the establishment of the following mode of drying or extend the drying process in the currently active mode.

The object of the study is the process of training artificial neural networks to identify the parameters of the materials creep curves.

The subject of the study is the parallelisation of artificial neural network training for the task of identifying the parameters of material creep curves using Python programming language, Chainer framework, and CUDA technology.

Purpose of the work is design and software implementation of the information system for artificial neural networks parallel learning algorithm for the task of identifying the parameters of the materials creep curves using Python programming language, Chainer framework, CUDA technology and cloud technologies.

The object of the study is the developing of an intelligent system for identifying the parameters of the materials

creep curves using artificial neural networks with a parallel learning algorithm.

The scientific novelty is the development of parallel algorithm for artificial neural networks learning using the Chainer framework, CUDA technology and cloud technologies.

The practical significance of the obtained results. The intelligent system which allows to determine the coefficients of mathematical models by identifying the parameters of materials creep was developed.

Analysis of using neural networks for identification parameters problem

The task of identification (processes, systems) or the construction of a mathematical model based on the results of observations occupies one of the main places in the modern theory of management and decision-making in various fields: engineering, economics, biology, etc. A separate and difficult question remains the definition of model parameters (parametric identification). In some cases they are known from the developer, in others they can be obtained as a result of measurements or tests with special effects on the control objects (identification experiments). However, most real systems are nonlinear in nature. Not only the parameters, but also the type of the model is often unknown to them. The only thing that is usually available in practice is the ability to communicate between system input and output. One of the promising methods use artificial neural networks that are universal approximation and allow with sufficient accuracy to simulate the dynamics of virtually any object [11].

The most effective mathematical models that can be used to predict the development of processes are those that use time series to construct them [22]. Traditional identification approaches are becoming less suitable for modelling complex nonlinear systems. Most processes cannot be described using traditional statistical models because they are substantially nonlinear and have either a chaotic or quasi-periodic or mixed (stochastic, chaos-dynamic, deterministic) basis. Adequate apparatus for building models of virtually any nonlinear structure can be methods that are based on artificial intelligence, namely artificial neural networks that have the ability to model nonlinear processes, adapt and allow you to work with noisy data. Radial-base structures that have high learning speeds and versatile approximation capabilities are promising [1], [6], [22].

Multi-layered, reverse error propagating neural networks are also widely used, which, with the backdrop of other intelligent tools that can be used to identify systems, have many significant benefits [1]. The general approach to solving the problem of identification in the context of artificial neural networks is shown in [6], but the question of practical implementation of parallel learning algorithms for network training in order to increase its efficiency remains unresolved.

The purpose of this article is to increase the efficiency of use of neural networks of direct distribution with back-propagation and their modifications in over-the cottages of parameter identification of creep of wood.

Comparative analysis of the study of training algorithms for artificial neural networks is performed by such criteria as the time of algorithm execution, the time of program implementation of algorithms, the speed of convergence of algorithms and the accuracy of the program result at a given

number of iterations. The run time of the algorithm is the number of steps that are performed in this algorithm from the beginning to the end. To count the number of steps investigated in this work algorithms, it is convenient to use the above block diagram [2], [4], [5], [8], [9], [10], [12], [13], [23], [24].

The number of steps characterizes nothing but the complexity of implementing algorithm, which is now a major factor, as often working hours for a programmer is more expensive than working hours on a computer, unlike in years past. The speed of program execution is generally influenced by the following factors:

- input information into the system;
- the quality of the compiled code of the program;
- time complexity of a program.

It should be noted that in case of complication of the network structure, the magnitude of the weight vector can become a factor that will have a significant impact on the learning time of ANN [2], [10]. In general, we can conclude that the optimal choice for ANN training in the problem of identification of rheological parameters of wood is the algorithm of back propagation of the error in view of the optimal ratio of learning time and accuracy. Such a parameter, as the complexity of the algorithm, plays a less important role, since in such a task, the accuracy of the obtained parameters is the main factor, since this accuracy is directly proportional to the results obtained from the implementation of mathematical models [5], [13], [18], [19].

Parallel calculations allow you to split the task into a number of smaller sub-tasks, but also to speed up the computations by performing parallel tasks and optimizing data sharing. The practical application and broad scope of tasks, which is only ever expanding, has led to an increase in the interest of the world's manufacturers of video accelerators to the possibilities of parallel computing and, as a consequence, the emergence of non-graphical general purpose calculations on GPUs (GPGPU, General Purpose computation on GPUs) [7], [14], [25]. GPU parallelization is achieved thanks to the structure of the GPU multiprocessor core, which operates on the principle of SIMD (single instruction, multiple data; one command stream, many data streams) and is, in fact, a parallel processor, allowing to perform special parallel algorithms for data processing.

One of such technologies the implementation of parallel non-graphical computations on GPUs is technology CUDA (Compute Unified Device Architecture) from the company NVIDIA, which has generated considerable interest in the global community. In particular [3], [7], [14], [25], [26] the wide spectrum of issues related to the implementation of programs using CUDA technology, and the creation of clusters on the basis of CPU i GPU. The emergence of non-graphical computing graphics cards paved the way for the development of new methods of parallel computing, not only on personal computers, since equipped with such a video card PC even without a multi-core processor can be used as a parallel computing machine of low power, but also to the emergence of new architectural solutions methods of organization of cluster computing.

From a technical point of view, there are several options of graphics processors, which greatly affect their computational power [7], [14]:

- number of computational cores with single and double precision (FP32 and FP64 Cores);

- the peak number of floating-point (single and double precision, Peak FP32 GFLOPs and Peak FP64 GFLOPs);
- Memory Interface
- bit and memory type (GDDR or HBM);
- Memory Size;
- size of second level cache (L2 Cache Size).

From a software perspective, you can classify GPUs based on the software interfaces (CUDA, OpenCL, OpenGL, and DirectX) and their maximum processor-supported version. In addition to these characteristics, the CUDA architecture provides capabilities that can greatly optimize the architecture.

The parallelization consists of two stages: the first is in the analysis of that task can be accomplished with the help of GPU computing capabilities, and the second stage is looking for all available GPUs with support for performing mathematical calculations, dividing the current task into sub-problems, what are the GPUs can perform in parallel, and pass these sub-tasks to the GPUs [26].

Parallelization at the level of the GPU is by creating special algorithms adapted to the specific architecture of graphics processors. At first glance, such a number of levels can increase the runtime due to the increased time of data transfer between levels and data exchange on the same level. Because the implementation of parallel computing at each level of architecture depends directly on the hardware on which the parallel algorithms will be implemented, each level of hybrid architecture has its own hardware and software. Today, the most progressive implementations of GPGPU technology is CUDA technology (Compute Unified Device Architecture), announced by NVIDIA in 2006 and supported on a number of NVIDIA graphics cards (GeForce 8, GeForce 9, GeForce 200, NVS, Quadro, and Tesla), and ATI Stream Technology (or ATI) FireStream and AMD Stream Processor) developed by AMD [25]. These prerequisites, as well as the high popularity of NVIDIA's video card solutions, have led to the choice of CUDA technology to parallelize the neural network learning algorithm and explore the possibility of more efficiently solving the tasks with the GPU.

Implementation of interaction between CUDA and Chainer framework. Using of AWS cloud technologies. Chainer provides implementation of training cycles implemented by the `chainer.training` module. It is based on many other basic functions Chainer, including Variable and Function, Link / Chain / ChainList, Optimizer, Dataset i Reporter Compared to the training cycle abstraction of other machine toolkits, the Chainer training base aims at maximum flexibility while maintaining simplicity for typical tasks [15].

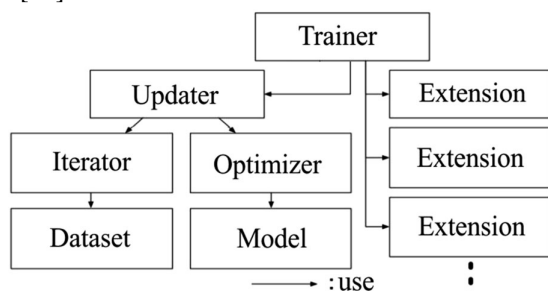


Fig. 1. Scheme implementation of Trainer in Chainer

Chainer provides a feature called Trainer that simplifies your model's learning process Figure 1. Writing your own

training cycle can be useful for learning how a trainer works or for implementing features not included in the standard Trainer. The core of the training cycle abstraction is Trainer, which implements the training cycle itself. The training cycle consists of two parts: Updater – which actually updates the settings for training, Extension – for arbitrary functions implemented in the learning algorithm.

Updater and some extensions use `chainer.dataset` and `Iterator` to scan datasets and download mini-batches. Trainer also uses `Reporter` to collect the observed values, and some extensions use `DictSummary` to collect and compute statistics. NVIDIA CUDA Deep Neural Network Library (cuDNN) is a graphics accelerating library of primitives for deep neural networks. cuDNN provides highly customizable implementation for standard procedures such as forward and backward convolutions, merging, rationing, and activation layers. cuDNN is part of the NVIDIA deep learning SDK [7], [14], [25], [26].

Researchers and developers rely on cuDNN to accelerate the GPU with high performance. This allows them to focus on learning neural networks and developing software applications, rather than spending time setting up low-GPU performance. cuDNN accelerates widely used deep learning frameworks, including Caffe2, MATLAB, Cognitive Toolkit, TensorFlow, Theano, and PyTorch. cuDNN is freely available to NVIDIA Developer Program members. It is advisable to use AWS to test the learning of neural networks using cloud technologies. Now the main task is to choose the EC2 machine. AWS enables you to select AMI to facilitate the machine setup process.

Amazon Machine Image (AMI) provides the information you need to run an instance that is a virtual server in the cloud. You can run multiple instances from the same AMI when you need multiple instances with the same configuration. You can use different AMIs to run instances when you need instances with different configurations. AMI may contain the operating system, required modules, tools, and dependencies.

Software implementation of the system

According to the complete learning procedure, first you need to prepare the dataset. The next step was the creation of the iterator on the dataset. Chainer provides some iterators that implement the generic strategy of creating mini-batches, repeating over data sets. `SerialIterator` is the simplest one that retrieves mini-batches in the main stream. `MultiprocessIterator` and `MultithreadIterator` is parallel version of `SerialIterator`. It supports operational sub-processes and sub-threads to load the next mini-batches in parallel.

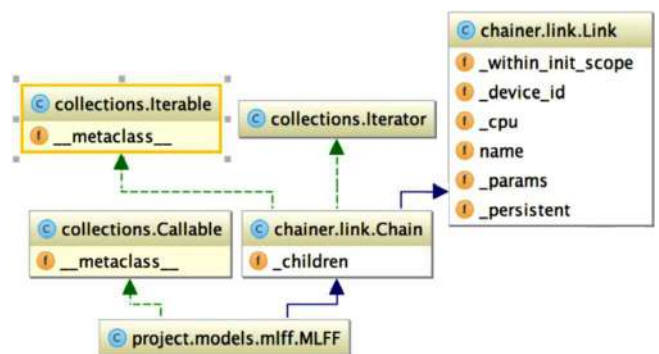


Fig. 2. Class diagram of the implementation of the multilayer neural network of direct distribution

Chain, Links and Functions have been implemented to implement the neural network model in Chainer. Chain is a combination of Links with an object interface (Figure 2).

An intelligent system for identification of wood creep parameters using a neural network with a parallel learning algorithm has been developed [15], [16], [17], [18], [19], [20], [21]. In particular, the diagram of variants of using of the training of the neural network is shown in Figure 3.

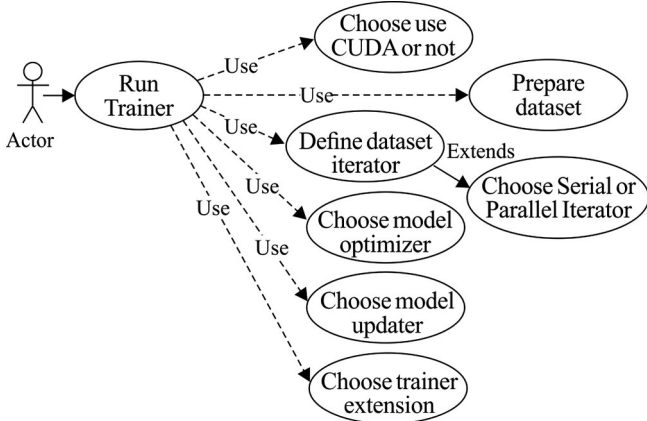


Fig. 3 Diagram of options for using the developed intellectual system

Composition is one of the most important features of neural networks. Neural networks are made up of many reusable fragments, and each model can be integrated into a larger training system. The network allows to describe the neural network on the basis of the composition such as collection parameters, serializing, copying the structure of the general parameters etc.

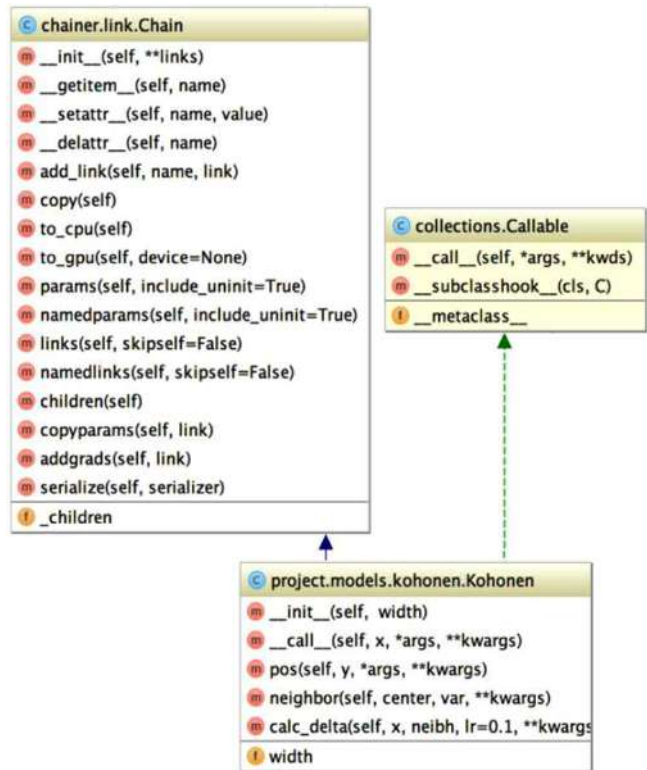


Fig. 4. Class diagram of the implementation of Kohonen neural network

Recurrent neural network training is implemented according to a typical training cycle, but Chainer Trainer does not have all the necessary attributes to train a Kohonen neural network, so some of them need to be implemented. The tra-

ining sample for training the network used is the same as for multi-layer neural network of direct distribution (Figure 4).

Application of artificial neural networks with parallel learning algorithm for identification of rheological parameters of wood. Determination of creep of wood is the starting point for the calculation of the basic rheological characteristics of the material during drying. For the study was using the principle of viscoelastic heredity. It, in turn, leads to the construction of integral equations.

For a linear stress state, the relationship between stresses $\sigma(t)$ and strains $\varepsilon(t)$ is described by integral equations

including creep $\varepsilon_{11}|_{ij}^n = \frac{u_{11}|_{ij+1}^n - u_{11}|_{ij}^n}{h_{x_1}}$ and relaxation

$\varepsilon_{22}|_{ij}^n = \frac{u_{22}|_{ij+1}^n - u_{22}|_{ij}^n}{h_{x_2}}$ kernels in the form [16], [17], [18], [19], [20], [21].

$$\varepsilon_{12}|_{ij}^n = \frac{1}{2} \left(\frac{u_{11}|_{ij+1}^n - u_{11}|_{ij}^n}{h_{x_2}} + \frac{u_{22}|_{i+1,j}^n - u_{22}|_{ij}^n}{h_{x_1}} \right), \quad (1)$$

$$\frac{\sigma_{11}|_{i+1,j}^n - \sigma_{11}|_{ij}^n}{h_{x_1}} + \frac{\sigma_{12}|_{ij+1}^n - \sigma_{12}|_{ij}^n}{h_{x_2}} = 0, \quad (2)$$

The creep function $\frac{\sigma_{22}|_{ij+1}^n - \sigma_{22}|_{ij}^n}{h_{x_2}} + \frac{\sigma_{12}|_{i+1,j}^n - \sigma_{12}|_{ij}^n}{h_{x_1}} = 0,$

which is the kernel of the integral equation, reflects the effect of a single voltage $\Pi(t - \tau)$, acting a single time interval at a time $R(t - \tau)$. The relaxation function

$\varepsilon(t) = \frac{\sigma(t)}{E} + \frac{1}{E} \int_0^t \Pi(t, \tau) \sigma(\tau) d\tau$ is the resolvent of kernel of the

creep $\sigma(t) = E\varepsilon(t) + E \int_0^t R(t, \tau) \varepsilon(\tau) d\tau$. We define the creep

function of wood as a linear combination of exponential operators.

$$\begin{aligned} \Pi &= \Pi(t, \tau) = \Pi_1(t - \tau_{rel}) + \Pi_2(\tau_{rel} - t) = \\ &= \frac{1}{\sigma_0} \left[\sum_{i=0}^{\infty} \omega_i e^{\nu_i(t - \tau_{rel})} \right] h(\tau) h(\tau_{rel} - t) + \\ &+ \frac{1}{\sigma_0} \left[\sum_{j=0}^{\infty} \eta_j e^{\mu_j(\tau_{rel} - t)} \right] h(t - \tau_{rel}), \end{aligned} \quad (3)$$

where, σ_0 is initial stresses; τ_{rel} - relaxation time; $\omega_i, \nu_i, \eta_j, \mu_j$ - parameters for which identification will be carried out.

Select the activation functions in the hidden layer of the neural network direct propagation in the form:

$$\phi_1(t) = \alpha_1 e^{\beta_1(\tau_{rel} - t)}, \quad (4)$$

$$\phi_2(t) = \alpha_2 e^{\beta_2(t - \tau_{rel})}, \quad (5)$$

So, get the handling mechanism of the rheological behavior of wood. By implementing the training algorithm of experimental data on creep, get identify rheological parameters of wood.

Figure 5 shows the approximation of experimental creep data of wood using an artificial neural network with the implementation of the parallelization of the learning algorithm by CUDA. The maximum deviation of the approximate values from the experimental values does not exceed 3 %.

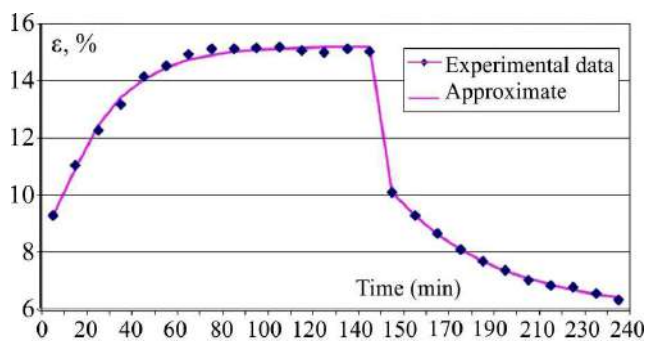


Fig. 5. The approximation of experimental data of creep of wood using artificial neural networks

Hence, it can be concluded that the use of direct propagation neural networks with activation functions selected in the form (4) i (5) is an effective tool for approximation of experimental creep data of wood and identification of parameters of mathematical models.

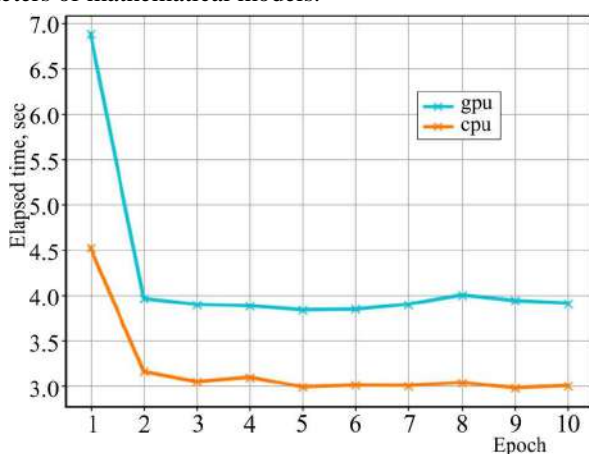


Fig. 6. Comparison of training time using 100 processes for parallel issue with and without using CUDA

The analysis of performance indicators shown in Figure 6 indicates the feasibility of using parallel learning algorithms for artificial neural networks, in particular, for the neural network Kohonen the learning time using parallelization using CUDA technologies was 4.51 sec., without using parallelization the learning time was 6.88 sec.

Conclusions

As a result of designing a system of parallel learning of artificial neural networks, an analysis of artificial intelligence systems and analysis of training of artificial neural networks were conducted. The structure of the Chainer framework and its interaction with CUDA is investigated. The analysis of existing cloud technologies for the implementation of the task and the analysis of algorithms for training artificial neural networks, their mathematical support; training algorithms were parallelized. Training on artificial neural networks using Amazon Web Services was conducted.

The necessary software of intelligent system of identification of creep parameters of wood using the neural network with parallel learning algorithm is developed. The analysis of performance indicators shown the feasibility of using parallel learning algorithms for artificial neural networks, in particular, for the neural network Kohonen the learning time using parallelization using CUDA technologies. Analyzing the results, we can conclude that the use of direct propagation neural networks with exponential activation functions on the hidden layer is an effective tool for approximation of experimental creep data and identification of parameters of mathematical models of wood rheological behavior.

imation of experimental creep data and identification of parameters of mathematical models of wood rheological behavior.

References

- [1] Anagnostopoulos, I., Anagnostopoulos, C., & Loumos, V. (2004). Classifying Web pages employing a probabilistic neural network. *IEEE Proceedings – Software*, 151(3), (pp. 139–150).
- [2] Bodyanskiy, Y. V., & Tyshchenko, O. K. (2019). A Hybrid Cascade Neuro-Fuzzy Network with Pools of Extended Neo-Fuzzy Neurons and its Deep Learning. *International Journal of Applied Mathematics and Computer Science*, 29(3), 477–488. <https://doi.org/10.2478/amcs-2019-0035>
- [3] Chang, D.-J., Kantardzic, M. M., & Ouyang, M. (2009). Hierarchical Clustering with CUDA/GPU. *ISCA PDCCS*, 7–12.
- [4] Chapman, B., Jost, G., & van der Pas, Ruud. (2008). Using OpenMP: portable shared memory parallel programming (Scientific and Engineering Computation), 2(3), 43–48. Cambridge, Massachusetts: The MIT Press.
- [5] Díaz, E., Brotons, V., & Tomás, R. (2018). Use of artificial neural networks to predict 3-D elastic settlement of foundations on soils with inclined bedrock. *Soils and Foundations*, 58(6), 1414–1422. <https://doi.org/10.1016/j.sandf.2018.08.001>
- [6] Gerbec, D., Gasperic, S., & Smon, I. (2005). Allocation of the load profiles to consumers using probabilistic neural networks. *IEEE Transactions on Power Systems*, 20(2), (pp. 548–555).
- [7] Gu, L., Li, X., & Siegel, Ja. (2010). An empirically tuned 2D and 3D FFT library on CUDA GPU. *Proceedings of the 24th ACM International Conference on Supercomputing – ACM*, Tsukuba, Japan. – June 01–04. New York, (pp. 305–314).
- [8] Haykin, S. (2013). *Neural Network a comprehensive foundation* (2nd ed.). Prentice Hall, 426 p.
- [9] Hong, S. G., Kim, S. W., & Lee, J. J. (2015). The Minimum Cost Path Finding Algorithm Using a Hopfield Type Neural Network. *Proceedings IEEE International Conference on Fuzzy, Systems 4*, (pp. 719–726).
- [10] Hu, Z., Bodyanskiy, Y., & Tyshchenko, O. K. (2019). Self-learning procedures for a kernel fuzzy clustering system. *Advances in Intelligent Systems and Computing*, 754, 487–497.
- [11] Spooner, J. T., Maggiore, M., Ordóñez, R., & Passino, K. M. (2002). *Stable Adaptive Control and Estimation for Nonlinear Systems: Neural and Fuzzy Approximator Techniques*. John Wiley & Sons, Inc., 236 p.
- [12] Krste, A., et al. (2016). The Landscape of Parallel Computing Research: A View from Berkeley University of California, Berkeley. *Technical cReport No. UCB/EECS-2016-183*.
- [13] Nabian, M. A., & Meidani, H. (2017). Deep Learning for Accelerated Reliability Analysis of Infrastructure Networks. *Computer-Aided Civil and Infrastructure Engineering*, 33(6), 443–458. <https://doi.org/10.1111/mice.12359>
- [14] Nukada, A., & Matsuoka, S. (2009). Auto-tuning 3-D FFT library for CUDA GPUs. *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis – ACM*, November 14–20, 2009, Portland, Oregon. – New York, (pp. 1–30).
- [15] Plas, D. V. (2018). *Python Data Science Handbook: Essential Tools for Working with Data*. St. Petersburg: Piter, 576 p.
- [16] Shymanskyi, V., & Protsyk, Yu. (2018). Simulation of the Heat Conduction Process in the Claydite-Block Construction with Taking Into Account the Fractal Structure of the Material. *Computer Science and Information Technologies: XIII-th International Scientific and Technical Conference (CSIT-2018)*, Lviv Ukraine, (pp. 151–154).
- [17] Sokolovskyy, Y., Boretska, I., Gayvas, B., Shymanskyi, V., & Gregus, M. (2019). Mathematical modeling of heat transfer in anisotropic biophysical materials, taking into account the phase transition boundary. *CEUR Workshop Proceedings 2488*, (pp. 121–132).

- [18] Sokolovskyy, Ya., Mokrytska, O., & Krishtapovich, V. (2015). Mathematical Simulation of Deformation and Relaxation process in capillary porous materials. *Proceedings of the information Conference on Computer Science and Information Technologies (CSIT 2015)*, Lviv Ukraine, (pp. 52–56).
- [19] Sokolovskyy, Ya., Nechepurenko, A., & Zdolbytskyi, A. (2017). Software simulation of heat mass transfer using parallel computing technologies. *Visnyk NULP: Komp'uterni systemy. Teoriia i praktyka*, 923, 34–43.
- [20] Sokolovskyy, Ya., & Sinkevych, O. (2018). Software and algorithmic support for representation of CAD models in 2D von Neumann neighborhood. *CEUR Workshop Proceedings 2300*, (pp. 215–218).
- [21] Sokolowskyi, Ya., Shymanskyi, V., & Levkovych, M. (2016). Mathematical modeling of non-isothermal moisture transfer and visco-elastic deformation in the materials with fractal structure. *Computer Science and Information Technologies. XI-Th International Scientific and Technical Conference (CSIT-2016)*, Lviv, Ukraine, (pp. 91–95).
- [22] Tian, B., Azimi-Sadjadi, M. R., & Vonder Haar, T. H. (2010). Temporal updating scheme for probabilistic neural network with application to satellite cloud classification. *IEEE Transactions on Neural Networks*, 11(3–4), (pp. 903–920).
- [23] Tkachenko, R., Izonin, I., Kryvinska, N., Chopyak, V., Lotoshynska, N., & Danylyuk, D. (2018). Piecewise-linear Approach for Medical Insurance Costs Prediction using SGTM Neural-Like Structure. In: Shakhovska, N., Montenegro, S., Estève, Ya., Subbotin, S., Kryvinska, N., Izonin, I. (Eds.): *Informatics & Data-Driven Medicine. Proceedings of the 1st International Workshop IDDM (IDDM 2018)*, Lviv, Ukraine, November 28–30, (pp. 170–179).
- [24] Tkachenko, R., & Izonin, I. (2019). Model and Principles for the Implementation of Neural-Like Structures based on Geometric Data Transformations. In: Hu, Z. B., Petoukhov, S., (Eds.): *Advances in Computer Science for Engineering and Education. Advances in Intelligent Systems and Computing. Springer, Cham. (ICCSEEA 2018)*. Vol. 754, (pp. 578–587).
- [25] Yang, C.-T., Huang, C.-L., & Lin, C.-F. (2011). Hybrid CUDA, OpenMP, and MPI parallel programming on multicore GPU clusters. *Computer physics communications*, 1, 266–269. <https://doi.org/10.1016/j.cpc.2010.06.035>
- [26] Zhiyi, Y., Zhu, Y., & Pu, Y. (2008). Parallel image processing based on CUDA. *Computer Science and Software Engineering*, 3, 198–201. <https://doi.org/10.1109/CSSE.2008.1448>

Я. І. Соколовський, В. М. Шиманський, О. В. Мокрицька, Я. В. Харко

Національний лісотехнічний університет України, м. Львів, Україна

НЕЙРОМЕРЕЖЕВА МОДЕЛЬ ІДЕНТИФІКАЦІЇ КРИВИХ ПОВЗУЧОСТІ МАТЕРІАЛУ З ВИКОРИСТАННЯМ CUDA ТЕХНОЛОГІЙ

Ця робота присвячена проблемі ідентифікації реологічних параметрів деревини з використанням штучних нейронних мереж з розпаралеленим алгоритмом навчання за допомогою мови програмування Python, фреймворку Chainer та технології CUDA. Розроблено інтелектуальну систему ідентифікації реологічних параметрів деревини. Створена система містить максимально зрозумілий інтерфейс користувача, весь необхідний комплекс інструментів для автоматизації процесу візуалізації та аналізу даних. У процесі створення інтелектуальної системи було передбачено вирішення наступних завдань: провести аналіз систем штучного інтелекту та аналіз навчання штучних нейронних мереж, зокрема багатоповзучих нейронних мереж прямого поширення, рекурентної нейронних мереж та нейронної мережі Кохонена; дослідити структуру фреймворку Chainer та його взаємодію з CUDA; провести аналіз існуючих хмарних технологій для реалізації завдання; провести аналіз алгоритмів навчання штучних нейронних мереж, їхнє математичне забезпечення; здійснити розпаралелення алгоритмів навчання та розробити необхідне програмне забезпечення. Використання Chainer дає змогу створювати пул пам'яті для розподілу пам'яті GPU. Щоб уникнути розподілу та вилучення пам'яті під час обчислень, Chainer надає можливість використовувати пул пам'яті CuPy як стандартний розподіл пам'яті не маючи справу з розподілом пам'яті. Для визначення фізико-механічних параметрів математичної моделі неізотермічного вологоперенесення та в'язко-пружного деформування капілярно-пористих матеріалів у процесі сушіння розроблено інтелектуальну систему. Вона надає можливість провести ідентифікацію параметрів ядер повзучості та релаксації, що записується як лінійна комбінація експоненціальних операторів. Запропонований алгоритм апроксимації та отримані розрахункові співвідношення реологічної поведінки деревини за допомогою багатоповзучої нейромережі з експоненціальними функціями активації у прихованих шарах дає змогу підвищити точність апроксимації експериментальних даних повзучості. Розроблені математичні моделі можуть бути використані для створення систем автоматизованого скінченно-різницевого розрахунку температури, вологовмісту та компонентів напружень під час сушіння капілярно-пористих матеріалів з урахуванням технологічних параметрів агента сушіння.

Ключові слова: ідентифікація параметрів; розпаралелювання; штучні нейронні мережі; python; chainer framework; cuda.

Інформація про авторів:

Соколовський Ярослав Іванович, д-р техн. наук, професор, завідувач кафедри інформаційних технологій.

E-mail: sokolowskyi@ukr.net; <https://orcid.org/0000-0003-4866-2575>

Шиманський Володимир Михайлович, канд. техн. наук, доцент, кафедра інформаційних технологій.

E-mail: vshymanskyi@gmail.com; <https://orcid.org/0000-0002-7100-3263>

Мокрицька Ольга Володимирівна, канд. техн. наук, доцент, кафедра інформаційних технологій.

E-mail: mokrytska@ntu.edu.ua; <https://orcid.org/0000-0002-2887-9585>

Харко Ярина Володимирівна, магістрант, кафедра інформаційних технологій. E-mail: kharkoaryna@ukr.net

Цитування за ДСТУ: Соколовський Я. І., Шиманський В. М., Мокрицька О. В., Харко Я. В. Нейромережева модель ідентифікації кривих повзучості матеріалу з використанням CUDA технологій. *Український журнал інформаційних технологій*. 2019, т. 1, № 1. С. 11–16.

Citation APA: Sokolovskyy, Ya. I., Shymanskyi, V. M., Mokrytska, O. V., & Kharko, Ya. V. (2019). Neural network model for identification of material creep curves using CUDA technologies. *Ukrainian Journal of Information Technology*, 1(1), 11–16. <https://doi.org/10.23939/ujit2019.01.011>