Section 2. Intelligent systems and data analysis

UDC 004.942
DOI https://doi.org/10.36059/978-966-397-538-2-8

MODELING OF NONLINEAR DYNAMICS USING THE SUPPORT MODEL METHOD

Dr.Sci. O. Fomin[0000-0002-8816-0652]**, Dr.Sci. S. Polozhaenko**[0000-0002-4082-8270]**,**
Ph.D. A. Savieliev[0000-0002-6949-5959]**, O. Tataryn**[0009-0005-3888-6569]
Odesa Polytechnic National University, Ukraine
EMAIL:fomin@op.edu.ua

Abstract. *This work is devoted to resolving the contradiction between the speed of constructing nonlinear dynamic models and their accuracy. The goal is to reduce the time required to build nonlinear dynamic models in the form of neural networks while ensuring the specified modeling accuracy. This goal is achieved by developing a modeling method based on the use of a set of pre-trained neural networks (support models) that reflect the main properties of the subject area. The scientific novelty of the developed method lies in the use of pre-trained neural networks with time delays as support models. Unlike existing pre-training methods, the developed method allows building simpler models with less training time while ensuring the specified accuracy. The practical benefit of the results of the work lies in the development of an algorithm for the support model method, which allows for a significant reduction in the training time of neural networks with time delays without losing modeling accuracy.*

Keywords: *identification; nonlinear dynamics; pre-training; neural networks, training speed.*

Introduction

The current stage of modelling development, which is mainly based on the use of intelligent technologies, is characterized by a number of requirements for both high accuracy of models and speed of their construction [1].

High accuracy in modelling nonlinear dynamics is currently achieved using machine learning methods, in particular neural networks (NN) [2, 3, 4]. However, the application of such methods is often associated with increased computational complexity, which leads to significant time cost on model construction [3, 4, 5].

The problem of reducing modelling time remains one of the most pressing, especially in areas related to the personalization of models that must adapt to changes in user behaviour or the environment (e.g., in authentication tasks, biomedical applications, human-machine systems), working in real time [6, 7].

Analysis of Previous Studies

One common approach to solving the problem of increasing the learning speed of neural networks is to optimize the architecture of models, which involves reducing the number of model parameters without significantly compromising their performance. Examples of such approaches include network simplification methods such as pruning, quantitation, and model compression [8, 9, 10].

Another relevant direction for increasing the training speed of NN is the use of accelerated learning algorithms, such as stochastic gradient descent with momentum (SGD with momentum) and adaptive optimization methods (e.g., Adam, RMSprop) [11, 12]. These methods make it possible to accelerate NN convergence thanks to an improved weight update strategy and a reduction in the number of epochs required to achieve the specified accuracy.

Another way to accelerate the NN training process is transfer learning [13, 14]. The main advantage of this method is the ability to use models pre-trained on a dataset from one subject area to solve target tasks from another subject area. This approach is used when the amount of specific data is limited or there are no high-quality datasets for the target task. Transfer learning is particularly effective when working with large neural networks, such as deep convolutional networks (CNNs), which require significant computational resources for training [15, 16]. One common approach to solving the problem of increasing the learning speed of neural networks is to optimise the architecture of models, which involves reducing the number of model parameters without significantly compromising their performance. Examples of such approaches include network simplification methods such as pruning, quantization, and model compression [17].

A special case of transfer learning can be considered preliminary training, in which the model is first trained on a large set of general data and then fine-tuned on more specific data for the target task [18, 19]. This approach significantly reduces the time and computational resources used during the fine-tuning of the target model, compared to training the model entirely on the target task data. As a result, pre-trained models converge faster and require fewer resources to achieve the desired model quality [18].

Transfer learning and pre-training technologies have become an integral part of many research and development projects in the field of artificial intelligence. They have proven their effectiveness in natural language processing tasks (BERT natural language processing network, GPT text generation network) [20], computer vision systems (pre-trained convolutional networks DenseNet, VGG) [17], biomedical research, and human-machine interfaces [20].

The widespread use of this approach to building models based on pre-training NN has been made possible by its practical advantages: a significant reduction in the cost of training models and an increase in their effectiveness in real-world applications, where not only accuracy but also speed is important. In addition, pre-trained models can be easily adapted to new tasks, making them indispensable tools in dynamically changing requirements and tasks.

This direction seems promising in the tasks of identifying nonlinear dynamic objects. At the same time, there is a lack of work in the field of preliminary training of neural networks that simulate the nonlinear dynamic properties of objects with continuous characteristics.

Based on the above, this paper develops an approach to building NN based on pre-training, which is capable of effectively coping with the requirements of modern modelling tasks, in particular the identification of nonlinear dynamic objects.

Unresolved Issues. The spread of the approach to building models based on prior training of neural networks has been made possible by its practical advantages: a significant reduction in the cost of training models and an increase in their effectiveness in real-world applications, where not only accuracy but also speed is important. In addition, pre-trained models can be easily adapted to new tasks, making them indispensable tools in dynamically changing requirements and tasks.

This direction also looks promising in the identification of nonlinear dynamic objects. At the same time, there is a lack of work in the field of pre-training NN that simulates the nonlinear dynamic properties of objects with continuous characteristics.

Based on the above, the paper develops an approach to building NN based on pre-training, which is capable of effectively coping with the requirements of modern modelling tasks, in particular the identification of nonlinear dynamic objects.

Problem statement

In practice, the use of prior training of neural networks involves considerable time expenditure on building a general model [14, 15]. To achieve the research goal, it is necessary to invent a method for accelerating the construction of general models of nonlinear dynamics. The formal formulation of the problem of accelerating the construction of general models of nonlinear dynamics based on prior training of neural networks is as follows. Let S be the domain for which there is enough labeling data N_S (D_S dataset):

$$D_S=\{(\mathbf{x}_i^S, y_i^S)\}, \quad (1)$$

where $\mathbf{x}_i^S$ is a vector of independent variables, y_i^S is the corresponding target variable (label), i=1,…, N_S.

Let $f_{\theta S}$ be the general NN model with θ_S parameters, which is trained on D_S dataset.

Let T be the target task in the subject area S, for which there are marked-up data of limited N_T size (D_T dataset):

$$D_T=\{(\mathbf{x}_j^T, y_j^T)\}, \quad (2)$$

where $\mathbf{x}_j^T$ is the vector of the independent variables, y_j^T is the corresponding target variable (label), j=1,…, N_T.

Let $f_{\theta T}$ be the target NN model with θ_T parameters, trained on D_T dataset, which ensures the accuracy $E_{\theta T}$ and the duration $t_{\theta T}$ of target model training.

It is necessary to find the parameters θ_S of the general model. Using them as starting values θ_{T0} during training, the target model $f_{\theta T}$, a specified accuracy level $E_{\theta T}$ is achieved in the minimum time $t_{\theta T}$:

$$\theta_{T0} = \theta_S : \underset{t_{\theta T}}{arg\ min} L_T(f_{\theta T}(\mathbf{x}_j^{\ T}), y_j^{\ T}) = E_{\theta T} \tag{3}$$

where L_T is the loss function adopted for the target model.

The objective of the article is to reduce the time required to build nonlinear dynamic continuous-time models in the form of neural networks while ensuring the specified modelling accuracy by developing a method based on prior training of neural network models.

To achieve this aim, the following tasks were formulated.

1. Development of a modelling method based on prior training by superimposing a set of pre-trained neural networks (support models) that reflect the main characteristics of the subject area.
2. Construction of support models in the form of neural networks that reflect the main nonlinear and dynamic characteristics of the subject area.
3. Investigation of the speed of modelling complex nonlinear dynamics using the developed support model method.

Support models method

The principle of identification based on support models. An approach to NN training based on the use of prior training in practice may result in a slight increase or, in general, a decrease in the training performance of the target model.

One of the factors reducing the productivity of the target model training process is the general nature and large volume of the training dataset D_{S0}, which must contain a description of the object's behavior in the widest possible range of external conditions and under the action of a wide range of input signals [66, 94]. As a result, in a specific case, when solving object modelling problems in a narrower range of external conditions and input influences, the coarse $f_{\theta S0}$ and accurate $f_{\theta Ti}$ models (i=1,…,g, where g is the number of target modelling problems) have excessive complexity (Fig. 1).

There are two ways to overcome this problem.

1. Forming a set of separate specialized training datasets D_{Si} (i=1,…,q, q – number of specialized training datasets) for building rough models $f_{\theta Si}$ with subsequent construction of accurate models $f_{\theta Ti}$ by further training the rough model on the target dataset D_T (Fig. 1b). In this case, the advantages of prior training are largely lost, since for each target task it is necessary to form a separate training dataset D_{Si} and train a rough model $f_{\theta Si}$. This leads to time losses when solving each target task.

The use of an approach based on the formation of separate specialized training datasets largely negates the advantages of prior training, since each new task requires the creation of a separate dataset and the training of a new model. This leads to significant time costs even at the task setting stage.

2. Formation of a set of separate training datasets D_{Sj} (j=1,…,h, where h is the number of basic characteristics reflecting the individual properties of the subject

area), to build support pre-trained neural networks $f_{\theta Sj}$ with subsequent construction of accurate models $f_{\theta Ti}$ by retraining a rough model in the form of a superposition of support pre-trained neural networks on the target dataset D_{Ti} (Fig. 1c).

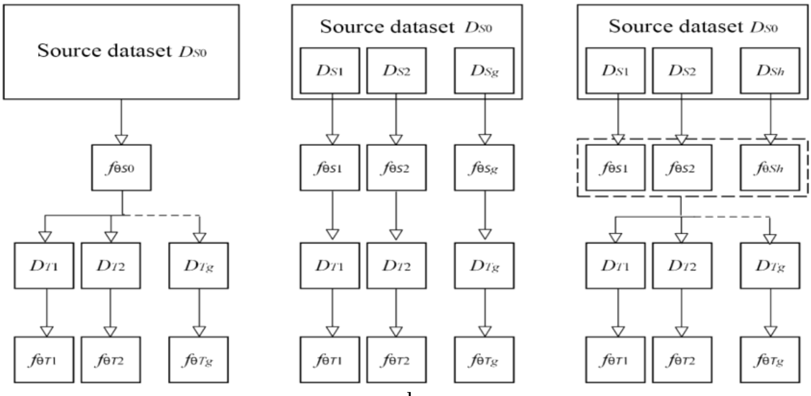


Figure 1. Structural diagram of the preliminary training process: a – preliminary training of a universal rough model; b – preliminary training of specific rough models; c – preliminary training of support models

The result of this approach is a set of support pre-trained NN $f_{\theta Sj}$, which are determined only once. At the same time, each of the basic NN $f_{\theta Sj}$ is significantly simpler than the rough model $f_{\theta S0}$.

The coarse model of the object $f_{\theta Si}$, which contains combined characteristics in the form of a superposition of typical dynamic and nonlinear links, is constructed on the basis of a set of support pre-trained NN $f_{\theta Sj}$ that correspond to the existing characteristics of the object. At the same time, the structure of the coarse model $f_{\theta Si}$ (the dimension of the parameter vector θ_{Si}) must coincide with the structure of the support NN $f_{\theta Sj}$ (the dimension of the parameter vector θ_{Sj}):

$$f_{\theta Si} : dim(\theta_{Si}) = dim(\theta_{Sj}), \tag{4}$$

which ensures the simplicity of the accurate model $f_{\theta Ti}$.

This approach assumes that a coarse model is built not on the basis of a lengthy preliminary training process, but on the basis of the superposition of several simple models (support models). Each support model describes a separate property of the subject area, and their superposition creates a refined coarse model.

The advantage of this approach is a significant increase in the speed of coarse model synthesis by eliminating the lengthy training phase of the coarse model for each new task in the subject area under consideration. In addition, the modularity of this approach makes it possible to use support models in different tasks, which reduces the time required to build models and provides flexibility and ease of replacement or improvement of individual support models.

The approach used in this work consists of using g support datasets D_{Sv} (v=1,…,g, g – the number of basic characteristics of the subject area), each of which describes a separate basic property of the area under study. Based on these datasets, support pre-trained models $f_{\theta Sv}(D_{Sv})$ with parameters θ_{Sv} are constructed. By combining and adapting the corresponding support models, a rough model $f_{\theta Sv}(D_{Sv})$ is constructed, which has a certain set of characteristics (non-linear and dynamic) of the object under study. The target model $f_{\theta Tk}(\theta_{Sv}, D_{Tk})$ of objects with certain characteristics is built by retraining the rough model on the D_{Tk} dataset.

The method of constructing complex models based on the superposition of simpler models is actively used in various scientific fields, such as theoretical physics, engineering, and, less frequently, in the fields of mathematical modelling and machine learning. Work in these fields shows that this method makes it possible to effectively solve complex system modelling problems, improving the accuracy and flexibility of models by dividing the problem into simpler subtasks. These methods are based on the principle of dividing a complex problem into subtasks, which simplifies the modelling process and improves the accuracy of solutions. Today, most of the efforts of researchers to solve the problems of identifying complex objects and improving the accuracy and speed of modelling are focused on the use of a rough model with subsequent retraining of an accurate model. Not enough attention is paid to the use of support models, although this approach seems promising due to its flexibility and modularity, which provides the following advantages.

1. Flexibility in development: Each support model is responsible for its own aspect of the system and can be easily replaced or improved without affecting the overall structure. This makes the development process modular and more manageable.

2. Simplification of complex tasks: Dividing a task into several subtasks with simplified models makes the process of modelling complex systems more understandable and easier to calibrate.

3. High accuracy: Synthesizing a target model from several base models allows for multiple aspects of the system to be taken into account, which ultimately leads to more accurate predictions.

The method of support models

To accelerate the construction of general models of nonlinear dynamics, a new approach is proposed in this paper. It consists in using a set of separate pre-trained NN (support models), each of which reflects separate basic characteristics of the area.

To construct a set of support models, a set of support datasets D_{Rk} is used (k=1,…,g, g is the number of basic characteristics of the domain). Each of the support datasets D_{Rk} describes a separate basic characteristic of the domain. On the basis of these datasets, support models $f_{\theta Rk}$ with the parameters θ_{Rk} are built. By combining support models that reflect the characteristics of the target problem, a general model $f_{\theta S}$ is built. It has a set of specified properties (nonlinear and dynamic). The target model $f_{\theta T}(\theta_S, D_T)$ is built by pre-training the general model $f_{\theta S}$ obtained on the basis of the set of support models $f_{\theta Rk}$ on the D_T dataset.

This approach allows to preserve the advantages of pre-training, since support models obtained once can be repeatedly used for different domains and target tasks, significantly reducing the total time and resources for training models without collecting additional data [17, 18, 19]. The structural scheme of the training process based on support models is presented in Fig. 2.

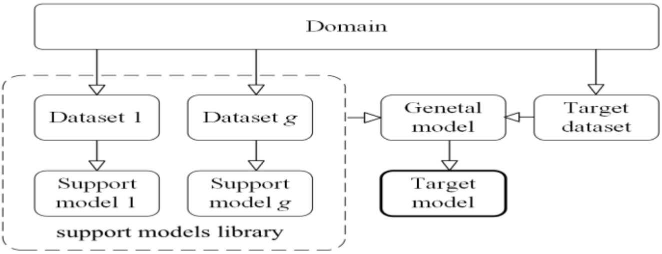


Figure 2. Structural diagram of the training process using support models

The algorithm of the suggested method consists of the following steps.

Step 1. Selection of basic domain properties and formation of a set of datasets D_{Rk} reflected the selected properties.

Step 2. Construction of support models set $f_{\theta Rk}$ in the form of separate NN corresponding to the established properties of the domain. Training of built models based on the generated datasets D_{Rk}.

Step 3. Determination of the list of p properties of the target problem from the set of g basic properties of the domain and construction of the general model $f_{\theta S}$ based on the superposition of the corresponding support models $f_{\theta Rh}$ (h=1,…,p, $p \leq g$) obtained in *Step* 2.

Step 4. Training of the target model $f_{\theta T}(\theta_S, D_T)$ based on the general model $f_{\theta S}$ obtained in *Step* 3.

Step 5. Determination of the accuracy indicators $E_{\theta T}$ and training time $t_{\theta T}$ of the target model $f_{\theta T}$. In case of unsatisfactory quality indicators of the target model $f_{\theta T}$, the control transfers to *Step* 2 to correct the structure θ_{Rk} of the support models $f_{\theta Rk}$, and, if necessary, to *Step 1* to correct the set of basic properties of the domain and the set of datasets D_{Rk}, reflecting the selected properties.

Selecting the basic properties of the domain for forming datasets

The basic properties of the domain in the work are understood as the characteristics of objects that reflect the essential features of their behavior. These properties can include real or abstract parameters that are important for solving the modeling problem [20, 21].

The procedure for selecting the basic properties of the domain and generating datasets D_{Rk} reflecting the selected properties is as follows.

1. Defining the range of tasks to be solved in a given domain; analyzing properties that are important for objects in a given domain, significantly affect the results of modeling and should be taken into account when forming the dataset D_S.

2. Determination of signal types (for example, periodic, random, pulsed) that best reflect the properties of the objects under study.

3. Formation of the dataset D_S based on the list of basic properties of the domain established in paragraph 1, the set of input signals and reactions of the object formed in paragraph 2.

4. Segmentation of the dataset D_S into separate datasets D_{Rk} according to the defined list of basic properties of the domain.

In the above sequence of steps, the task of determining the type of signals that best reflect the properties of the object under study remains the least formalized. Automating the selection of support models is crucial for scaling the method to complex, multidimensional dynamical systems and bridges the gap between human intuition and automated processing, allowing the application of machine learning methods for objective identification and segmentation.

Determination of the structure of support models and their pre-training

To modelling nonlinear dynamics, the work uses time-delayed NN (TDNN) [21, 22, 23]. Due to their simplicity and versatility, TDNNs are most widely used in modeling problems of nonlinear dynamic objects. In practice, the TDNN structure is most often used, consisting of three layers: input, hidden, and output [22]. The size of the layers in this TDNN structure is determined as follows: the input layer consists of M neurons and is responsible for the memory (dynamic characteristics) of the model; the hidden layer consists of K neurons and is responsible for the nonlinear characteristics of the model; the output layer contains the number of Y neurons, which is equal to the number of outputs of the model.

For each support model, a labeled data set $D_{Si}=\{(x^{Pk}_{Hj}(t), f_{Vi}[x^{Pk}_{Hj}(t)])\}$ is formed based on the signals $x(t)$ at the input of the object and the responses $y(t)= f_{Vi}[x(t)]$ at its output. Typical signals are often used as input signals: impulse $x(t)=a\delta(t)$, step $x(t)=a\Theta(t)$, linear $x(t)=at$, and harmonic $x(t)=a\ sin(t)$ signals of various amplitudes $a \in (0, 1]$. Time delays at the input are implemented through shifts in time series and the inclusion of previous values in the input vector.

The main steps for implementing time delays and forming a training sample are as follows:

Step 1. Selecting the number of delays. Determine the number of delays M that will be used.

Step 2. Forming the input vector. For each time moment t_k, the input vector is formed as a sequence of current and previous values.

Step 3. Transferring delays to the network. Each formed input vector is transmitted to the input layer of the neural network, where it is processed as a regular input.

Construction of a general model based on the superposition of the corresponding support models

After completing the process of pre-training the set of support models $f_{\theta Rk}$, a general model $f_{\theta R}$ is built on their basis. This model is composed of a set of *p* support models $f_{\theta Rh}$ that correspond to the available basic properties of the object. The selection of support models that reflect the basic properties of an object is generally subjective. To reduce subjectivity and increase the reproducibility of the selection of support models, it is advisable to use methods of clustering input data or feature contribution analysis.

After completing the preliminary training process for the family of support models that correspond to the existing basic characteristics of the object, a rough model $f_{\theta S}(D_S)$ is constructed based on them. Assuming that the general $f_{\theta S}$ and support $f_{\theta Rk}$ models are constructed in the form of NN with the same structure (dimension of the parameter vector $dim(\theta_S)=dim(\theta_{Rk})$), the definition of the general model is reduced to calculating the arithmetic operations on corresponding components of the parameter vectors of the support models θ_{Rk}.

At the same time, several approaches to the superposition of support models are considered:

additive superposition is used when each support model is responsible for independent aspects of the system (e.g., dynamics and environmental impact). The outputs of the support models $f\theta Si(DSi)$ are determined based on the calculation of the arithmetic mean of the corresponding components of the parameter vectors of the support models θSv:

$$\theta_S^i = \frac{1}{h}\sum_{k=1}^{h}\theta_{Rk}^i \tag{5}$$

where *i* are the indices of the corresponding elements of the parameter's vectors of the general θ_S and the support θ_{Rk} models.

multiplicative superposition is used in the case of interacting processes (e.g., where some processes modify others). In this case, the outputs of the support models $f_{\theta Sv}(D_{Sv})$ are multiplied:

$$\theta_S^i = \prod_{v=1}^{b}\theta_{Sv}^i \tag{6}$$

combined superposition methods use complex combinations of support models, such as weighted sums of outputs of nonlinear functions to combine the outputs of several models, the application of nonlinear functions to the outputs of individual models, etc. Such methods include determining the parameter vector θ_S of the approximate model as the maximum value among the corresponding components of the parameter vectors of the support models θ_{Sv}:

$$\theta_S^i = max(\theta_{Sv}^i), v = \overline{1,b} \tag{7}$$

Thus, the advantage of forming an approximate model using the support models method is the absence of a training procedure, which reduces computational complexity and significantly speeds up the process of building an approximate model. At the same time, the dimension of the approximate model $f_{\theta S}$ (the dimension of the parameter vector θ_S) remains the same as in the base models, i.e., the complexity of the approximate model does not increase. Another advantage of forming a general model using expressions (5)-(7) is the absence of a training procedure, which significantly speeds up the process of constructing a approximate model.

Experiment setup

The study of the support models method is carried out on the example of a nonlinear dynamics test object. A simulation model of a test object in the form of a sequence of a nonlinear link with saturation and a dynamic link of the first order [24, 25] is shown in Fig. 3.

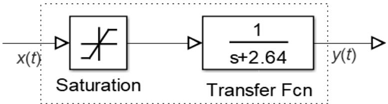


Figure 3. Test object simulation model

As typical characteristics from the set of properties of the domain of nonlinear dynamics, a nonlinear characteristic in the form of saturation and a dynamic link of the first order were chosen to describe the behavior of the test object. For the test object, a labeled D_S dataset generated on the base of signals $x(t)$ at the input of the object and the responses $y(t)$ at its output. The inputs are pulsed $x(t)=a\delta(t)$, stepped $x(t)=a\Theta(t)$, linear $x(t)=at$ and harmonic $x(t)=a\cdot sin(t)$ signals of different amplitudes $a \in (0, 1)$. Based on the dataset D_S, two support datasets are formed:

input stepped signals $x(t)=a\Theta(t)$ and responses $y(t)$ of the object with nonlinearity in the form of saturation D_{R1};

input stepped signals $x(t)=a\Theta(t)$ and responses $y(t)$ of the object in the form of a dynamic link of the first order D_{R2}.

The experiment consists in studying the training speed of the target model of the test object, built by various methods:

based on the general model $f_{\theta S}(D_S)$, pre-trained on the general D_S dataset;

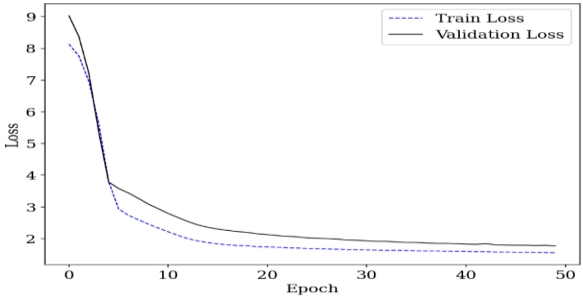
based on individual support models $f_{\theta R1}(D_{R1})$, $f_{\theta R2}(D_{R2})$, pre-trained on datasets D_{R1} and D_{R2};

based on the general model $f_{\theta R}(f_{\theta R1}, f_{\theta R2})$ in the form of a superposition of support models $f_{\theta R1}(D_{R1})$ and $f_{\theta R2}(D_{R2})$.

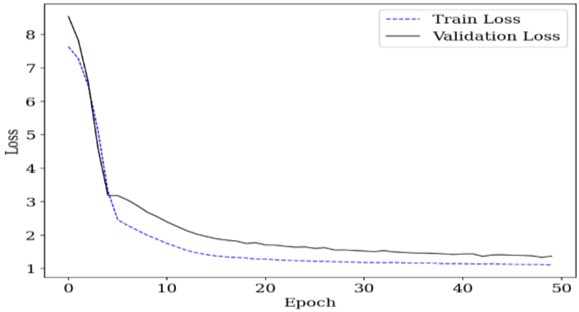
Simulation and results

Building a rough model. To determine the structure of the model $f_{\theta S}(D_S)$, which is a three-layer TDNN, based on the results of additional research, the number of neurons M=30 in the input layer and K=30 in the hidden layer was accepted. The model is trained using the backpropagation method with network parameter updating using the Levenberg-Marquardt method. Preliminary training is limited to 50 epochs to prevent overfitting and preserve adaptability.

Fig. 4 shows the dependence of loss functions (*mse, mae*) during training of the coarse model on the number of training epochs.



a



b

Figure 4. Dependence of loss functions during training of a rough model on the number of training epochs: *a* – *mse*, *b* – *mae*

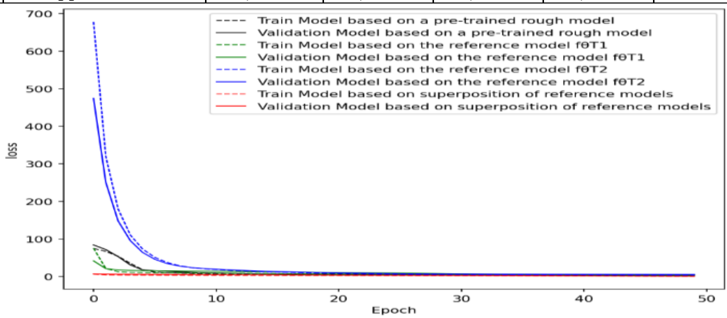
Construction of an accurate model. The structure of the accurate model $f_{\theta T}(\theta_S, D_T)$ based on the pre-trained coarse model $f_{\theta S}(D_S)$, models $f_{\theta Tk}(D_{Tk})$ based on separate support models of datasets D_{T1} and D_{T2}, and model $f_{\theta T}(D_T)$ based on the superposition of support models is selected to be identical to the coarse model $f_{\theta S}(D_S)$ in the form of a three-layer TDNN. The NN is trained using the backpropagation method with network parameter updating using the Levenberg-Marquardt method. Training of the accurate model is carried out over 50 epochs.

Fig. 5 shows the dependence of loss functions (*mse*, *mae*) on the number of training epochs for accurate models based on the coarse model, based on separate support models $f_{\theta T1}(D_{T1})$ and $f_{\theta T2}(D_{T2})$, and based on the superposition of support models. The results of the experiment on studying the training speed of the accurate model of the test object, built on the basis of a rough model, on the basis of separate support models, and on the basis of a superposition of support models, are given in Table 1.

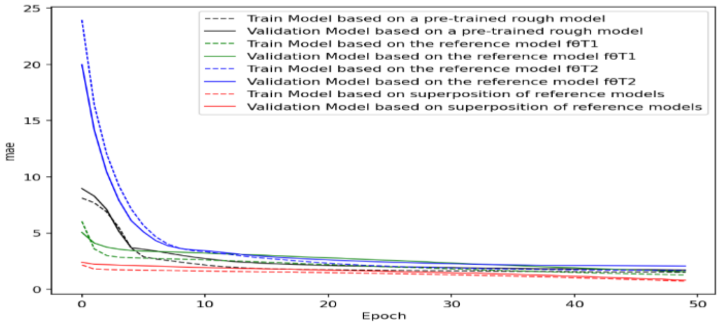
Table 4

Results of an experiment to study the learning speed of a target model of a test object

No	Model based on	Training		Validation		Number of epoch
		mse	*mae*	*mse*	*mae*	
1	general model	4,9105	1,8890	6,9738	2,3972	14
		2,2267	1,2686	2,9774	1,4809	50
2	support model (saturation)	4,9596	1,8623	6,8963	2,4457	8
		2,7553	1,4094	4,1863	1,7621	50
3	support model (first-order dynamic link)	4,8058	1,8544	6,8876	2,2571	23
		3,9412	1,6869	5,7969	2,0801	50
4	superposition of support models	4,5837	1,8347	6,6431	2,2256	3
		0,9254	0,7941	1,0704	0,8807	50



a



b

Figure 5. Dependencies of loss functions during training of accurate models based on a coarse model, separate support models, and superposition of support models on the number of training epochs: *a – mse*, *b – mae.*

Fig. 5 and Table 1 show the advantage of using pre-trained support NN during the identification of nonlinear dynamic objects, namely, a significant reduction in the training time of the TDNN model (4.6 times) compared to the traditional approach of building an accurate model based on a pre-trained coarse model with comparable accuracy of both models (established quality requirements at the validation stage: *mse*=7.0, *mae*=2.5). The use of individual support models as coarse models can also reduce the training time of the accurate model (by 1.8 times) with comparable accuracy of both models.

Conclusions and prospects

The paper successfully solves the problem of reducing the time of building nonlinear dynamics continuous-time models in the form of neural networks while ensuring the specified accuracy of modeling. To resolve the conflict between the accuracy of modeling nonlinear dynamic objects and the speed of model construction, a modeling method was developed based on pre-training through the superposition of support models that reflect the basic properties of the subject area.

The effectiveness of the developed method for modeling nonlinear dynamics was proven when solving the problem of modelling a test nonlinear dynamic object. The experiment demonstrates a 4.6-fold reduction in the time of building a target model using support models compared to the traditional modeling method based on pre-training. The advantages of the proposed approach are the ability to quickly adapt to changing operating conditions, high speed of building the target model while ensuring the specified modeling accuracy. In addition, the developed method allows improving the efficiency of model training in the lack of labeled data for the target task. The disadvantages of the proposed approach, inherited from methods based on pre-training, are the dependence of the modeling results on the quantity and quality of data of the target dataset.

The practical limitations of the application of the proposed method are the a priori need for support models built on a sufficient amount of qualitative data. Insufficient data or poor data quality can significantly reduce the accuracy of support models and, as a result, significantly reduce the training time of an accurate model.

Thus, the area of effective application of the proposed method is allocated: lack of marked data of the target task in the presence of a general dataset of sufficient size; no significant discrepancies between the characteristics of the general and target datasets.

To improve and expand the scope of application of the support models method, it is necessary to take into account the real conditions of the external environment by expanding the experimental part at different levels of noise distortion and under conditions of time drift of the observed parameters. In order to fully assess the potential of the proposed method and determine its place among advanced solutions in further research, it is planned to expand the range of test objects, including systems with different dynamics, multidimensional systems, objects with delays, etc.

References

1. Chisty, N. M. A., & Adusumalli, H. P. (2022). Applications of artificial intelligence in quality assurance and assurance of productivity. *ABC Journal of Advanced Research, 11*(1), 23–32. https://doi.org/10.18034/abcjar.v11i1.625

2. Sen, J. (2021). *Machine learning – algorithms, models and applications*. IntechOpen. https://doi.org/10.5772/intechopen.94615

3. Li, M. (2025). The past decade and future of the cross application of artificial intelligence and decision support system. In *2025 IEEE 6th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT)* (pp. 1666–1669). IEEE. https://doi.org/10.1109/AINIT65432.2025.11036058

4. Chitty-Venkata, K. T., Emani, M., Vishwanath, V., & Somani, A. K. (2023). Neural architecture search benchmarks: Insights and survey. *IEEE Access, 11*, 25217–25236. https://doi.org/10.1109/ACCESS.2023.3253818

5. Adikari, I., Lakmali, S., Dhananjaya, P., & Herath, D. (2023). Accuracy comparison between recurrent neural networks and statistical methods for temperature forecasting. In *2023 3rd International Conference on Advanced Research in Computing (ICARC)* (pp. 72–77). IEEE. https://doi.org/10.1109/ICARC57651.2023.10145620

6. Kariri, E., Louati, H., Louati, A., & Masmoudi, F. (2023). Exploring the advancements and future research directions of artificial neural networks: A text mining approach. *Applied Sciences, 13*(3), 3186. https://doi.org/10.3390/app13053186

7. Sinha, N. K., Gupta, M. M., & Rao, D. H. (2000). Dynamic neural networks: An overview. In *Proceedings of IEEE International Conference on Industrial Technology 2000* (pp. 491–496). IEEE. https://doi.org/10.1109/ICIT.2000.854201

8. Kutyniok, G. (2022). The mathematics of artificial intelligence. *Proceedings of the International Congress of Mathematicians, 7*, 5118–5139. https://doi.org/10.4171/ICM2022/141

9. Kästner, C., & Kang, E. (2020). Teaching software engineering for AI-enabled systems. In *The 42nd International Conference on Software Engineering (ICSE). Software Engineering Education and Training*. https://arxiv.org/abs/2001.06691

10. Gholizade, M., Soltanizadeh, H., Rahmanimanesh, M., et al. (2025). A review of recent advances and strategies in transfer learning. *International Journal of System Assurance Engineering and Management, 16*, 1123–1162. https://doi.org/10.1007/s13198-024-02684-2

11. Hosna, A., Merry, E., Gyalmo, J., et al. (2022). Transfer learning: A friendly introduction. *Journal of Big Data, 9*(1), 102. https://doi.org/10.1186/s40537-022-00652-w

12. Zhuang, F., Qi, Z., Duan, K., et al. (2020). A comprehensive survey on transfer learning. *Proceedings of the IEEE, 99*(9), 1–34. https://doi.org/10.1109/JPROC.2020.3004555

13. Jones, I., Swan, J., & Giansiracusa, J. (2024). Algebraic dynamical systems in machine learning. *Applied Categorical Structures, 32*(4). https://doi.org/10.1007/s10485-023-09762-9

14. Xu, W., He, J., & Shu, Y. (2020). Transfer learning and deep domain adaptation. In *Advances and Applications in Deep Learning*. IntechOpen. https://doi.org/10.5772/intechopen.94072

15. Lu, Y., Jiang, X., Fang, Y., & Shi, C. (2021). Learning to pre-train graph neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence, 35*(5), 10. https://doi.org/10.1609/aaai.v35i5.16552

16. Siebert, J., Joeckel, L., Heidrich, J., et al. (2022). Construction of a quality model for machine learning systems. *Software Quality Journal, 30*, 307–335. https://doi.org/10.1007/s11219-021-09557-y

17. Karampiperis, P., Manouselis, N., & Trafalis, T. B. (2002). Architecture selection for neural networks. In *Proceedings of the 2002 International Joint Conference on Neural Networks (IJCNN'02)* (Vol. 2, pp. 1115–1119). IEEE. https://doi.org/10.1109/IJCNN.2002.1007650

18. Pal, K., & Patel, B. V. (2020). Data classification with K-fold cross validation and holdout accuracy estimation methods with 5 different machine learning techniques. In *Fourth International Conference on Computing Methodologies and Communication (ICCMC)* (pp. 83–87). IEEE. https://doi.org/10.1109/ICCMC48092.2020.ICCMC-00016

19. Nakamichi, K., et al. (2020). Requirements-driven method to determine quality characteristics and measurements for machine learning software and its evaluation. In *2020 IEEE 28th International Requirements Engineering Conference (RE)* (pp. 260–270). IEEE. https://doi.org/10.1109/RE48521.2020.00036

20. Liao, T. W. (2005). Clustering of time series data—a survey. *Pattern Recognition, 38*(11), 1857–1874. https://doi.org/10.1016/j.patcog.2005.01.025

21. Leylaz, G., Wang, S., & Sun, J. Q. (2022). Identification of nonlinear dynamical systems with time delay. *International Journal of Dynamics and Control, 10*, 13–24. https://doi.org/10.1007/s40435-021-00783-7

22. Fomin, O. O., & Orlov, A. A. (2024). Modeling nonlinear dynamic objects using pre-trained time delay neural networks. *Applied Aspects of Information Technology*, *7*(1), 24–33. https://doi.org/10.15276/aait.07.2024.2

23. Wenyuan, L., Zhu, L., Feng, F., et al. (2020). A time delay neural network based technique for nonlinear microwave device modelling. *Micromachines, 11*(9), 831. https://doi.org/10.3390/mi11090831

24. Fomin, O., Arsenyeva, O., Romanova, T., Sukhonos, M., & Tsegelnyk, Y. (2022). Interpretation of dynamic models based on neural networks in the form of integral-power series. In *Smart Technologies in Urban Engineering* (Lecture Notes in Networks and Systems, Vol. 536, pp. 258–265). Springer. https://doi.org/10.1007/978-3-031-20141-7_24

25. Fomin, O. O., Ruban, O. D., & Rudkovskyi, O. V. (2020). Method for construction the diagnostic features space of switched reluctance motors based on

integral dynamic models. *Problemele energeticii regionale, 4,* 35–44. https://doi.org/10.5281/zenodo.4316968

МОДЕЛЮВАННЯ НЕЛІНІЙНОЇ ДИНАМІКИ МЕТОДОМ ОПОРНИХ МОДЕЛЕЙ

Dr.Sci. О. Фомін[0000-0002-8816-0652]**, Dr.Sci. С. Положаєгко** [0000-0002-4082-8270]**,**
Ph.D. А. Савєльєв[0000-0002-6949-5959]**, О. Татарин**[0009-0005-3888-6569]
Національний університет «Одеська політехніка», Україна
EMAIL: fomin@op.edu.ua

Анотація. *Робота присвячена вирішенню суперечності між швидкістю побудови нелінійних динамічних моделей та їх точністю. Метою роботи є скорочення часу побудови моделей нелінійної динаміки у вигляді нейронних мереж при забезпеченні заданої точності моделювання. Ця мета досягається шляхом розробки методу моделювання, заснованого на використанні набору попередньо навчених нейронних мереж (опорних моделей), що відображають основні властивості предметної області. Наукова новизна розробленого методу полягає у використанні в якості опорних моделей попередньо навчених нейронних мереж з тимчасовими затримками. На відміну від існуючих методів попереднього навчання, розроблений метод дозволяє будувати більш прості моделі з меншим часом навчання при забезпеченні заданої точності. Практична користь результатів роботи полягає в розробці алгоритму методу опорних моделей, який дозволяє істотно скоротити час навчання нейронних мереж з тимчасовими затримками без втрати точності моделювання.*

Ключові слова: *моделювання; нелінійна динаміка; попереднє навчання; нейронні мережі з часовими затримками.*

UDC 004.048
DOI https://doi.org/10.36059/978-966-397-538-2-9

PROCESSING PIPELINE FOR ASTRONOMICAL DATA MINING USING AI-BASED DECISION-MAKING RULES

Ph.D. S. Khlamov1[0000-0001-9434-1081], **Dr.Sci. V. Savanevych**2[0000-0001-8840-8278],
Yu. Netrebin 3[0009-0001-8778-3241], **T. Trunova**4[0000-0003-2689-2679],
I. Tabakova 5[0000-0001-6629-4927]
Kharkiv National University of Radio Electronics, Ukraine
EMAIL: [1]sergii.khlamov@gmail.com, [2]vadym.savanevych1@nure.ua,
[3]yurii.netrebin@nure.ua, [4]tetiana.trunova@nure.ua,
[5]iryna.tabakova@nure.ua