

Time Series Prediction based on Hybrid Neural Networks

S.A. Yarushev¹, A. V. Fedotova^{2,*},
V.B. Tarasov², A.N. Averkin³

[*afedotova.bmstu@gmail.com](mailto:afedotova.bmstu@gmail.com)

¹Dubna State University, Dubna, Russia

²Bauman Moscow State Technical University, Moscow, Russia

³Institution of Russian Academy of Sciences Dorodnicyn
Computing Centre of RAS, Moscow, Russia

In this paper, we suggest to use hybrid approach to time series forecasting problem. In first part of paper, we create a literature review of time series forecasting methods based on hybrid neural networks and neuro-fuzzy approaches. Hybrid neural networks especially effective for specific types of applications such as forecasting or classification problem, in contrast to traditional monolithic neural networks. These classes of problems include problems with different characteristics in different modules. The main part of paper create a detailed overview of hybrid networks benefits, its architectures and performance under traditional neural networks. Hybrid neural networks models for time series forecasting are discussed in the paper. Experiments with modular neural networks are given.

Keywords: time series, forecasting, modular neural networks, hybrid neural networks

Introduction

Nowadays modelling and forecasting time series are among the most active areas of research. For example, depending on the historical data, situation on sales market, changes in prices for shares of population growth and banks deposits are forecast. Forecasting time series affects the lives of people around the world, so it has great practical value and perspectives of research in all areas of the modern society, which is also an important area in the field of computer application.

The solution of problems of identification of dynamic objects should be used in a variety of fields: it can simplify temperature controllers, or complex management and forecasting. It can also solve the forecasting problem, along with a number of different methods, for example, statistical analysis, neural networks [1]. Identification of the object may be difficult if the exact structure of the model of the object is unknown, some of the parameters of the object change due to obscure principles, or the exact number of parameters of the object is unknown. In such cases, the hybrid neural network can be used for identification of dynamic objects. There are many types of neural networks that are used for identification of dynamic objects. Despite the large

number of neural network methods for identification of dynamic objects, most of these algorithms have some limits, or do not provide the required accuracy.

Among all kinds of neural networks, architectures that can be used for identification of dynamic objects allocated a class of neural networks based on self-organizing maps of Kohonen with hybrid architecture. Hybrid neural networks of this type will get special attention in this article because they are becoming more widespread and successful applications for solving various problems of recognition [2], identification [3], and forecasting. We will also consider a number of biomorphic neural networks applicable for solving identification problems and management.

1. Modular Neural Networks

1.1. Main idea of the modularity

The core of the modular neural networks is based on the principle of decomposition of complex tasks into simpler ones. Separate modules make simple tasks. More simple subtasks are then carried through a series of special models. Each local model performs its own version of the problem according to its characteristics. The decision of the integrated object is achieved by combining the individual results of specialized local computer systems in a dependent task. The expansion of the overall problem into simpler subtasks can be either soft or hard-unit subdivision. In the first case, two or more subtasks of local computer systems can simultaneously assigned while in the latter case, only one local computing model is responsible for each of the tasks crushed.

Each modular system has a number of special modules that are working in small main tasks. Each module has the following characteristics [4]:

- The domain modules are specific and have specialized computational architectures to recognize and respond to certain subsets of the overall task;
- Each module is typically independent of other modules in its functioning and does not influence or become influenced by other modules;
- The modules generally have a simpler architecture as compared to the system as a whole. Thus, a module can respond to given input faster than a complex monolithic system;
- The responses of the individual modules are simple and have to combine by some integrating mechanism in order to generate the complex overall system response.

The best example of modular system is human visual system. In this system, different modules are responsible for special tasks, like a motion detection, color recognition and shape. The central nervous system, upon receiving responses of the individual modules, develops a complete realization of the object which was processed by the visual system.

1.2. Artificial Hybrid Neural Networks

Definition 1: A neural network is hybrid if it has a set of subsystems operating in parallel independently of each other and has different outputs which are indirectly integrated and do not interact with other.

Hybrid neural networks are especially effective for specific types of applications such as forecasting or classification problem, in contrast to traditional monolithic neural networks. These classes of problems include problems with different characteristics in different modules. For example, in the case of function approximation, piecewise continuous function does not model conventional neural networks, at the same time, hybrid neural networks solve this problem quite effectively. [5] Some of the main advantages of hybrid training systems are scalability, additional training, constant adaptation, economics of education and training, as well as computational efficiency.

Hybrid neural networks are made up of subsystems that may be separated according to categories based on different structures and functionalities; each subsystem is combined with one another. Each subsystem may represent a separate neural network, which performs individual subtasks. Different learning algorithms can also be combined with each other, which leads to better training the neural network by integrating the best learning algorithms for a particular task. Structurally, the hybridity of the a priori knowledge problem can be introduced into the structure of a neural network that provides significant structural representation of it. Typically, different approaches to hybridization used in combination with each other in order to achieve an optimum combination of a hybrid network structure and a learning algorithm.

1.3. Advantages of Hybrid Neural Networks

1) Simplification of the traditional neural network system

The complexity of monolithic neural networks greatly increases the complexity and size of the problem. The number of scales quadratically increases in relation to the size of most neural networks [6]. Hybrid neural networks can avoid this problem as a special subsystem (modules) splits the problem into simpler and solves them [7], [8].

2) Immunity

Homogeneous communication of traditional neural networks leads to poor stability and susceptibility to interference. Hybrid neural networks increase reliability and fault tolerance models. Such properties have been observed in the structure of the visual system of the brain, which has a modular design and consists of separate, independent modules, which are interconnected. Damage to one of the modules is not able to destroy the whole entire system, it will operate [9].

3) Extensibility

Scalability is one of the most important features of the hybrid neural networks. Monolithic neural networks, if necessary retraining, it is necessary to train fully again. While a hybrid neural networks need not be completely retrained due to its design, there is a possibility of adding new modules, and if it is necessary, retrain a separate module.

4) There is no need to retrain the entire network

Hybrid neural networks are the basis of integration, capable of controlled and uncontrolled learning paradigms. The modules can be pre-trained individually for specific subtasks and then integrated by the integration unit, or can be trained along with the integrating unit. In the later

situation, there is no indication of training data, a module must perform subtasks, and that during the preparation of the individual modules to compete or cooperate to achieve the desired overall objectives. This training scheme is a combined function of both controlled and uncontrolled learning paradigms.

5) Efficiency

The division into more simple tasks helps to significantly reduce the computational cost [10]. The hybrid neural network can learn a variety of functional maps faster than the corresponding monolithic global neural network, because each individual module in a hybrid neural network must learn perhaps more simple, functional part of the general mapping. Additionally, hybrid network have the inherent ability to degrade the degradable tasks simpler set of tasks, thereby enhancing the Learn-ability and training time.

6) Easy to learn

Embedding hybrid neural networks leads to a large number of advantages compared to a single global neural network. For example, implementation of complex modes of local area networks of neurons improves learning ability hybrid neural network model, and thus makes them more complicated for large-scale problems that usually cannot processed by the global neural network models. Furthermore, complex behavior may require different types of equipment and process knowledge to be combine with each other, which is possible without any structural or functional hybridity.

7) Speed training

To ensure the continued survival of biological systems, new functionalities integrated into existing systems, along with the continuation of learning and adapting to changing conditions [11]. Similar, the hybridity enables learning economy so that in case of changing operating conditions, only those parts of the hybrid neural network must be change, which does not correspond to the new conditions, and not the entire system. Furthermore, it is possible also to reuse some of the existing specialized modules for different applications of the same nature, rather than re-training details common to the two problems.

8) Integrality

Hybridity is a way of embedding a priori knowledge in a neural network architecture, which is important to improve the learning of the neural network. The motivation for the integration of a priori knowledge of the problem is that it might be the best way to develop an appropriate neural network system for the available training data. This may include the ability to hybridize the neural network architecture. The hybrid neural network architecture can be use to integrate the various neural functions, various neural structures or various kinds of learning algorithms, depending on the task.

9) Insight into Neural Network Models

The hybrid neural networks can achieve significant performance improvements as well as knowledge about the task can be use to introduce the structure and significant representation in their design. By virtue of the fact that different modules perform different tasks within separate hybrid neural network, and the intermediary unit adjusts its behavior, easily get an idea of the

workings of hybrid neural network simply by separately analyzing the output behavior of the individual modules and the intermediary unit. This function does not exist, and perhaps not even possible in global monolithic neural networks.

10) Biological analogy

Hybrid or modular neural networks have an analogy with biological nervous systems, which operate on a similar principle. The nervous system is, in turn, from the various subsystems that solve their problems, that work together to achieve global targets the nervous system. For example, a very complex task Detect broke down into smaller sub-tasks, so that the visual tasks optimized for different situations. In addition, the same structure could itself replicate many times, giving the visual cortex, a much-desired property of robustness.

2. Hybrid Neural Networks Models for Time Series Forecasting

2.1. VQTAM - Vector-Quantized Temporal Associative Memory.

VQTAM - a modification of the self-organizing Kohonen maps, which could be used for identification of dynamic systems [11]. The structure of this network is analogous-tech structure of Kohonen self-organizing maps, the key difference lies in the organization of the weighting coefficients network of neurons. The vector of input features of the network is divided into two parts: $\mathbf{x}^{in}(t)$, $\mathbf{x}^{out}(t)$. The first part of the input features $\mathbf{x}^{in}(t)$ contains information about the inputs of a dynamic object and its previous output. The second part of the vector of input features $\mathbf{x}^{out}(t)$ contains information about the alleged withdrawal of the dynamic object corresponding inputs. Weight vector is split in a similar manner. Thus,

$$\mathbf{x}(t) = \begin{pmatrix} \mathbf{x}^{in}(t) \\ \mathbf{x}^{out}(t) \end{pmatrix} \text{ and } \mathbf{w}_i(t) = \begin{pmatrix} \mathbf{w}_i^{in}(t) \\ \mathbf{w}_i^{out}(t) \end{pmatrix},$$

where $\mathbf{w}_i(t)$ - weight vector of i -neuron, $\mathbf{w}_i^{in}(t)$ weights of part of the vector corresponding to the part of the input vector $\mathbf{x}^{in}(t)$, and $\mathbf{w}_i^{out}(t)$ - part of the vector of weights corresponding to the portion of the input vector $\mathbf{x}^{out}(t)$. The first part of the vector of input features contain information about process inputs and outputs of its previous:

$$\mathbf{x}^{in}(t) = (\mathbf{y}(t-1), \dots, \mathbf{y}(t-n_y), \mathbf{u}(t), \mathbf{u}(t-1), \dots, \mathbf{u}(t-n_u)),$$

where $n_y \ll T, n_u \ll T$. The second part of the vector of input features

$$\mathbf{x}^{out}(t) = \mathbf{y}(t)$$

contains information about the intended output of this process, the appropriate inputs $\mathbf{x}^{in}(t)$.

Each example is a sample consisting of a pair of vectors

$$(\mathbf{y}(t), \mathbf{u}(t))$$

and sampling should withhold not less than

$$\max(n_u, n_y)$$

examples. The vector $\mathbf{y}(t)$ is a vector of the object at time t inputs and $\mathbf{u}(t)$ - outputs the vector of the object at the same time.

After the input of the next input vector $\mathbf{x}(t)$ network is made up of not-how many examples of learning sample, the neuron-winner is determined only by the vector $\mathbf{x}^{in}(t)$:

$$i^*(t) = \arg \min_i \left\{ \left\| \mathbf{x}^{in}(t) - \mathbf{w}_i^{in}(t) \right\| \right\},$$

where $i^*(t)$ - number of neuron-winner in step t .

To change the weights the modified rule changes weights for conventional SOM could be applied:

$$\Delta \mathbf{w}_i^{in}(t) = \alpha(t) h(i^*, i, t) [\mathbf{x}^{in}(t) - \mathbf{w}_i^{in}(t)],$$

$$\Delta \mathbf{w}_i^{out}(t) = \alpha(t) h(i^*, i, t) [\mathbf{x}^{out}(t) - \mathbf{w}_i^{out}(t)],$$

where $0 < \alpha(t) < 1$ - the speed of the network training, and h - the function of the neuron i and i^* . The neighbourhood $h(i^*, i, t)$. As a neighbourhood $h(i^*, i, t)$ function a Gaussian function, for example, could be selected:

$$h(i^*, i, t) = \exp \left(- \frac{\left\| \mathbf{r}_i(t) - \mathbf{r}_{i^*}(t) \right\|^2}{2\sigma^2(t)} \right),$$

$\mathbf{r}_i(t)$ and $\mathbf{r}_{i^*}(t)$ wherein - the position on the map i and i^* neurons, respectively, $\sigma(t) > 0$ - one to radius-defined neighbourhood function in step (usually the initial value of the selected parameter decreases linearly or exponentially depending on). After the selection of the winning neuron network output is set $\mathbf{w}_{i^*}^{out}(t)$ (of-trained network diagram is displayed in Fig. 1). Where TDL - Tapped Delay Line.

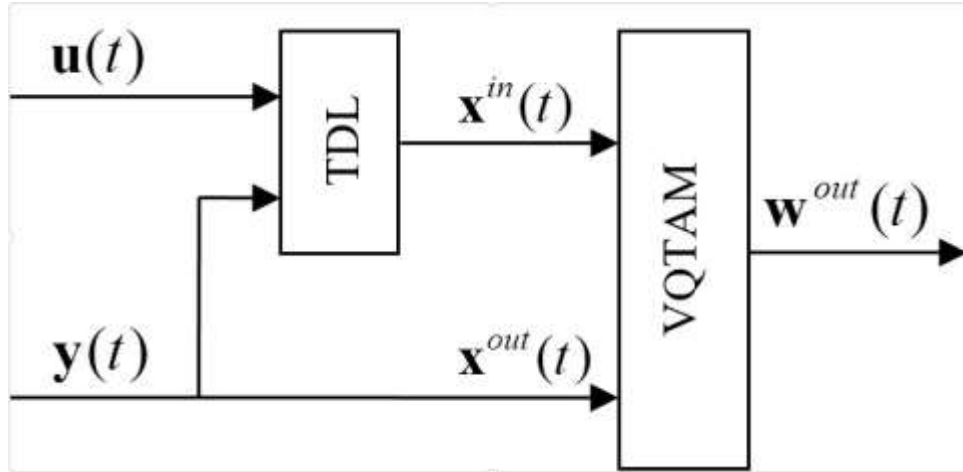


Fig.1. Learning scheme of VQTAM.

While working on VQTAM input is only the vector $\mathbf{x}^{in}(t)$ for which determined by the neuron-winner and the output is set to the network $\mathbf{w}_{i^*}^{out}(t)$. The vector $\mathbf{w}_{i^*}^{out}(t)$ may be a life interpreted as predicted-yield of the $\hat{\mathbf{y}}(t)$ object at the time t . It is also worth noting that this algorithm is applied to objects, the output of which can be described as a means of continuous and discontinuous functions.

2.2. RSOM – Recurrent Self-Organizing MapVQTAM - Vector-Quantized Temporal Associative Memory

In RSOM, unlike ordinary Kohonen maps with recurrent connections, a decaying in time vector of outputs is introduced for each neuron. This vector is used to determine the winner neuron and in maps weights modifications [12].

We represent the vector network inputs as follows:

$$\mathbf{x}(t) = (\mathbf{y}(t-1), \dots, \mathbf{y}(t-n_y), \mathbf{u}(t), \mathbf{u}(t-1), \dots, \mathbf{u}(t-n_u)) ,$$

where $n_y \ll T$, $n_u \ll T$.

The output of each neuron is determined by the following expression:

$$V_i(t) = \|\mathbf{v}_i(t)\| ,$$

where $\mathbf{v}_i(t) = (1 - \alpha)\mathbf{v}_i(t-1) + \alpha(\mathbf{x}(t) - \mathbf{w}_i(t))$ - constant output attenuation coefficient ($0 < \alpha \leq 1$), $V_i(t)$ - i th neuron output in cycle t , $\mathbf{w}_i(t)$ - i th weight vector of the neuron, $i = \overline{1, k}$, k - the number of neurons in the network.

After filing for another example of the network, input neuron-winner is determined as a neuron with minimum output:

$$i^*(t) = \arg \min_i \{V_i(t)\} .$$

To change the scale a modified rule for training Kohonen maps is used:

$$\Delta \mathbf{w}_i(t) = \alpha(t)h(i^*, i, t)\mathbf{v}_i(t) ,$$

where $0 < \alpha(t) < 1$ - the speed of the network training, and h - the function of the neuron i and i^* the neighbourhood. Driving such a neuron network shown in Fig. 2.

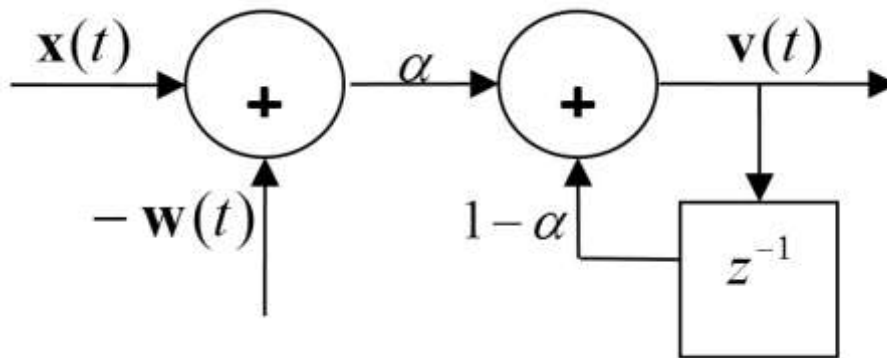


Fig.2: Neuron scheme of RSOM network.

After the learning network - the network is launched on the training set and the cluster-polarizes it, forming clusters that could be approximated by local models such as linear functions. Local model is built for each cluster. Thus, after running a test sample on the network for each sampling point determined by the most suitable local model in which the following output object can be predicted.

This process could be accelerated by using the algorithms of construction of local linear models during training of the neural network. This method of construction of models is applicable, if the dimension of the vector output of the object is equal to one, it is discussed further scalar values $y(t)$ and $\hat{y}(t)$ so it is worth noting that this method can be successfully applied only to objects, the output of which can be described by a continuous function. Each RSOM neuron network assigned a matrix $A_i(t)$ containing the coefficients of the corresponding linear model:

$$A_i(t) = [\mathbf{b}_{i,1}(t), \dots, \mathbf{b}_{i,n_y}(t), \mathbf{a}_{i,1}(t), \dots, \mathbf{a}_{i,n_U}(t)]^T$$

The output value of the network is determined in accordance with the following expression:

$$\hat{y}(t) = \sum_{k=1}^{n_y} \mathbf{b}_{i^*,k}(t) \mathbf{u}(t-k) + \sum_{l=1}^{n_U} \mathbf{a}_{i^*,l}(t) \mathbf{y}(t-l) = A_{i^*}^T(t) \mathbf{x}(t)$$

where $A_{i^*}(t)$ - coefficient matrix associated with neuron winner $i^*(t)$. The Matrix $A_{i^*}(t)$ is used for linear approximation of the model output.

In the construction of local linear models simultaneously with the neural network training, there must be an additional rule to change the coefficients of linear models:

$$A_i(t+1) = A_i(t) + \alpha' h(i^*, i, t) \Delta A_i(t)$$

where $0 < \alpha' < 1$ - the speed of learning model, and $\Delta A_i(t)$ - usually error correction Widrow-Hoff:

$$\Delta A_i(t) = [y(t) - A_i^T(t) \mathbf{x}(t)] \frac{\mathbf{x}(t)}{\|\mathbf{x}(t)\|^2},$$

where $y(t)$ - the desired output of the model when applied to the input vector $\mathbf{x}(t)$.

Thus, a change in the model coefficients with weight coefficients of neurons in each network training step. Online mode after applying the input vector input - neuron network determines the winner $i^*(t)$. Then the corresponding matrix $A_{i^*}(t)$ coefficients of the linear model are selected. With the help of selected matrix output model is defined by the formula: $\hat{y}(t) = A_{i^*}^T(t) \mathbf{x}(t)$.

2.3. Modular Self-Organising Maps

Modular self-organizing maps are presented in a number of works by Tetsuo Furukawa [14]. SOM has a modular structure of the array consisting of mo-functional modules, which are trained neural network, for example, multi-layer perceptrons (MLP), and not the vectors as in the conventional self-organizing maps. In the case of MLP-modules, modular self-organizing map identifies the group of features or functions, depending on the input and output values, while building a map of their similarity. Thus, modular self-organizing map with MLP modules pre-constitutes a self-organizing map in a functional, rather than in a vector-space [15].

Such neural network structure can be regarded as biomorphic, as they emerged, a novena is largely due to research the structure of the cerebral cortex with milk-feed, and confirmed by a number of further studies [16]. The basis of the idea of the structure of the cerebral cortex is the model of a cellular structure where each cell is a plurality of neurons, a neural column. Column neurons are combined into a more complex structure. In this regard, it is proposed to simulate the column neuro-new separate neural networks. This idea has formed the basis of modular neural networks.

In fact, a modular self-organizing map is a common Kohonen map where neurons are replaced by more complex and independent structures, such as other neural networks. This change requires a slight modification of algorithm training.

In the proposed algorithm Furukawa at the initial stage of training network repents on the sample input data corresponding to different images (take to the problem to be solved in this work - the different states of a dynamic object), which can map the similarities to build a network, and calculate the error of each of the network module:

$$E_i^k = \frac{1}{J} \sum_{j=1}^J \|\hat{\mathbf{y}}_{i,j}^k - \mathbf{y}_{i,j}\|^2.$$

Module-winner is determined as a module, minimizing the error E_i^k :

$$k_i^* = \arg \min_k E_i^k.$$

Once the module-winner is determined, the network adapts the scales – at first the weight of the winning module is adapted according to one of the possible learning algorithms of this type of networks then begins adapting weight cards. In this process, the parameters of each of the modules are treated as a card weight and are adapted for standard algorithms Kohonen self-organizing maps.

Articles such Furukawa neural network is used to construct maps of different similarity, interpolation, and pattern recognition. In this present work, modular networks have been successfully applied to the identification of dynamic objects, with unique, previously proposed modular structures have been developed, where the modules and the type of network VQTAM RSOM described above are used.

Modular neural network, where the modules used type of network VQTAM (ie SOMxVQTAM Network), developed in the course of this work, is trained using a combination of learning algorithms modular network, as well as learning algorithm VQTAM types of networks. At the stage of training, after the training of the next example, network input, calculated outputs VQTAM all types of networks are used as modules, module-winner is further defined as a minimum unit whose output is removed from the expected net yield on the submitted training example. After that there is a network adjustment weights VQTAM type, located in the winning unit, adjustment of the scale where the algorithm adjusts the weights VQTAM types of networks. The set of weight vector of the module of neural network is considered as one of the weight vectors of the modular network. And the weight modification is done according to the standard of Kohonen Maps learning algorithms.

In the case of the network module where the modules as applied RSOM type network (i.e. network SOMxRSOM) are also obtained in this work, learning occurs in a similar manner.

Next, we consider some examples of work; produce new neural networks in a number of real-world data and compare them with some other algorithms.

Conclusion

This work considered in detail the main features of the construction of the hybrid neural network models for time series prediction. In particular, we consider the advantages of hybrid architectures over traditional monolithic neural networks.

The paper presents several predictive models developed on the basis of hybrid modular neural network and presents their main characteristics.

Further work on the topic of the research suggests a deeper consideration of the topic of forecasting with the use of hybrid systems in conjunction modular architectures and fuzzy systems.

Acknowledgements

The work was supported by Russian Foundation for Basic Research, projects № 14-07-00603 and № 16-37-50023.

References

1. Haykin S. Neural networks: a comprehensive foundation. *Macmillan*. New York, 1994.
2. Efremova N., Asakura N., Inui T. Natural object recognition with the view-invariant neural network. In: *5th International Conference of Cognitive Science*, 2012, pp. 802-804.
3. Trofimov, A., Povidalo I. And Chernetsov S. Usage of the self-learning neural networks for the blood glucose level of patients with diabetes mellitus type 1 identification. *Science and education*, 2010, vol. 5. Available at: <http://technomag.edu.ru/doc/142908.html>, accessed 18.12.2016.
4. Haykin S. Neural networks: a comprehensive foundation. *Macmillan*. New York, 1994.
5. Perugini N., Engeler W.E. Neural network learning time: Effects of network and training set size. In: *International Joint conference on neural networks*, 2: 1989, pp. 395-401.
6. Gomi H., Kawato M. Recognition of manipulated objects by motor learning with hybrid architecture networks. *Neural Networks*, 6: 1993, pp. 485–497.
7. Azam F., Vanlandingham H.F. A hybrid neural network method for robust handwritten character recognition. In: *Artificial Neural Networks for Intelligent Engineering, ANNIE'98*, 1998, vol. 8, pp. 503-508.

8. Lee T. Structure level adaptation for artificial neural networks. *Kluwer Academic Publishers*, 1991.
9. Kosslyn S. Image and Brain. *MIT Press*, Massachusetts, 1994.
10. Stork B. Non-optimality via pre-adaptation in simple neural systems, In: *Artificial Life II, Proceedings of the Workshop on Artificial Life*. Held February, 1990, Santa Fe, New Mexico, 1991, vol. 3, pp. 409-429.
11. French R. Catastrophic forgetting in connectionist networks. In: *Trends in Cognitive Sciences*, 3(4): 1999, pp. 128-135.
12. Gustavo L., Souza M, Barreto A. Multiple Local ARX Modeling for System Identification Using the Self-Organizing Map. In: *II European Symposium on Time Series Prediction*, 2008, pp. 215-224.
13. Koskela T. Neural network methods in analyzing and modelling time varying processes. *Helsinki University of Technology*, Espoo, 2003. pp. 1-72.
14. Tokunaga K., Furukawa T. SOM of SOMs. *Neural Networks*. 2009, vol.22, pp. 463-478.
15. Tokunaga K., Furukawa T. Hybrid network SOM. *Neural Networks*. 2008, №22, pp. 82-90.
16. Vetter T., Hurlbert A., Poggio T. View-based Models of 3D Object Recognition: Invariance to Imaging Transformations. *Cerebral Cortex*. 1995, vol. 3, pp. 261-269.

УДК 004.415.2

Прогнозирование временных рядов на основе гибридных нейронных сетей

Ярушев С. А.¹, Федотова А. В.^{2,*},
Тарасов В. Б.², Аверкин А. Н.³

* afedotova.bmstu@gmail.com

¹Государственный Университет «Дубна», Дубна, Россия

²МГТУ им. Н.Э. Баумана, Москва, Россия

³Вычислительный центр им. А.А. Дородницына РАН Федерального
исследовательского центра «Информатика и управление» РАН,
Москва, Россия

Ключевые слова: временные ряды, прогнозирование, модулярные нейронные сети, гибридные нейронные сети

Прогнозирование временных рядов представляет собой обширную область, которая развивается наиболее быстрыми темпами. Способствует всему этому быстрое изменение ситуации во всех областях жизни, это и экономика, и политика, и различные другие сферы, которые непосредственно влияют на жизнь каждого из нас. Методы прогнозирования также эволюционируют в след за изменяющейся конъюнктурой и предъявляемыми новыми и новыми требованиями. Изменяются производственные и экономические процессы, меняется законодательная база, которая регулирует данные процессы, появляются новые процессы производственной и социальной сферы, все это влечет за собой появление физически коротких временных рядов, поскольку данные процессы, индикаторы не являлись предметом статистического учета. Также, трудности для прогнозирования представляют нелинейные процессы, зашумленность временных рядов. Исходя из данной ситуации целесообразно разрабатывать новые методы прогнозирования, а как показывают исследования, лучше всего с этим справляются гибридные методы.

Идея модулярных нейронных сетей основывается на принципе декомпозиции сложных задач на более простые. Эта идея схожа с тем, как построена биологическая нервная система, которая обладает очень важным свойством – при выходе из строя одного из модулей, другие продолжают работать исправно. Благодаря построению гибридных модульных нейросетевых систем, можно получать универсальные и устойчивые системы.

Гибридные и модульные архитектуры нейронных сетей обладают широким рядом преимуществ над традиционными нейронными сетями. Среди них, можно выделить способность данных сетей к расширению без необходимости переобучения всей нейронной сети, что доставляет не мало проблем разработчикам. Достаточно переобучить один мо-

дуль и сеть может работать. Гибридные сети гораздо более стабильны к помехам, они гораздо быстрее обучаются, а процесс обучения проще. Это только часть характеристик, детально изложенных в данной работе.

Предложены несколько модификаций модульных нейронных сетей, на основе самоорганизующихся карт Кохонена, такие как Vector-Quantized Temporal Associative Memory (VQTAM), Recurrent SOM (RSOM), Modular SOM. Детально описана архитектура данных нейронных сетей.

Работа выполнена при финансовой поддержке РФФИ по проектам № 16-37-50023 и 14-07-00603.

Список литературы

1. Haykin S. Neural networks: a comprehensive foundation. New York: Macmillan, 1994.
2. Efremova, N., Asakura N., Inui T.. Natural object recognition with the view-invariant neural network. In: 5th International Conference of Cognitive Science, 2012, pp.802-804.
3. Trofimov, A., Povidalo I. And Chernetsov S. Usage of the self-learning neural networks for the blood glucose level of patients with diabetes mellitus type 1 identification. Science and education, 2010, vol. 5. Available at: <http://technomag.edu.ru/doc/142908.html>
4. Haykin S. Neural networks: a comprehensive foundation. New York : Macmillan, 1994.
5. Perugini N. And Engeler W. E. Neural network learning time: Effects of network and training set size. In: International Joint conference on neural networks, 2: 1989, pp.395–401.
6. Gomi H. And Kawato M. Recognition of manipulated objects by motor learning with hybrid architecture networks. Neural Networks, 6: 1993, pp.485–497.
7. Azam F. And Vanlandingham H. F. A hybrid neural network method for robust handwritten character recognition. In Artificial Neural Networks for Intelligent Engineer-ing, ANNIE'98, 1998, vol. 8, pp. 503–508.
8. Lee T. Structure level adaptation for artificial neural networks. Kluwer Academic Publishers, 1991.
9. Kosslyn S. Image and Brain. MIT Press, Massachusits, 1994.
10. Stork B. Non-optimality via pre-adaptation in simple neural systems, In: Artificial Life II, Proceedings of the Workshop on Artificial Life.Held February, 1990, Santa Fe, New Mexico, 1991,vol. 3, pp. 409–429.
11. French R. Catastrophic forgetting in connectionist networks. In: Trends in Cognitive Sciences, 3(4): 1999, pp. 128–135.
12. Gustavo L., Souza M, Barreto A. Multiple Local ARX Modeling for System Identification Using the Self-Organizing Map. In: II European Symposium on Time Series Prediction, 2008, pp. 215-224.
13. Koskela T. Neural network methods in analyzing and modelling time varying processes – Espoo, 2003. pp. 1–72.
14. Tokunaga K., Furukawa T. SOM of SOMs. Neural Networks. 2009, vol.22, pp. 463 – 478.

15. Tokunaga K., Furukawa T. Hybrid network SOM. Neural Networks. 2008, №22, pp. 82–90.
16. Vetter T., Hurlbert A., Poggio T. View-based Models of 3D Object Recognition: Invariance to Imaging Transformations. Cerebral Cortex. 1995, vol. 3, pp. 261 – 269.