Development of Stability Control Mechanisms in Neural Network Forecasting Systems

I. Miloserdov and D. Miloserdov
St. Petersburg Institute for Informatics and Automation of Russian Academy of Sciences (SPIIRAS), 39, 14 Line, St. Petersburg 199178, Russia
dmnl94@yandex.ru

Abstract. The problem of ensuring the stable functioning of time series forecasting systems based on streaming recurrent neural networks with controlled elements is considered. The mechanisms necessary and sufficient for its maintenance are derived, which involve maintaining the balance of the learning history and modifying the synapse learning rules in order to establish a balance between positive and negative potential. The results of experiments to assess the accuracy of forecasting are presented.

1. Introduction
Both direct propagation neural networks [1] and recurrent neural networks (RNN) are used for time series forecasting [2, 3]. Hybrid solutions are known that combine different neural network architectures or combine neural networks and traditional methods [4]. RNN allow for deeper information processing. They have mechanisms for associative call of information from their memory. These neural networks include RNN with controlled elements [5]. Unlike their counterparts, these RNN can be endowed with different space-time structures, which increases their potential for time series forecasting.

At the same time, despite the known advantages, the fundamental problem of the neural network stability remains largely unexplored. Training of neural network is an iterative process, and the network itself is a dynamic system with feedback. Neural network architectures and training methods are quite diverse, which makes it difficult to synthesize universal algorithms for stability analysis. Recently a lot of investigations have been devoted to this issue. In [6], the model of an artificial Hopfield neural network with non-monotonic nonlinear connections is considered. In [7], the method for analyzing the stability of feedback systems with neural network controllers is presented. For Hopfield neural networks, the Mittag-Leffler stability criterion was obtained in [8]. Inertial neural networks were studied in [9, 10]. Thus, sufficient conditions for global asymptotic stability are obtained in [9], and conditions for exponential stability in the sense of Lagrange are obtained in [10]. Sufficient stability conditions for Hopfield reaction-diffusion networks are considered in [11].

The paper is devoted to the mechanisms for ensuring the stability of neural network systems for time series forecasting, which are based on RNNs with controlled elements. These systems were used in [12, 13] and demonstrated an advantage over known solutions, but due to the lack of mechanisms for monitoring the stability of functioning, the operator's participation was required in the operation of such systems.
The aim of the study is to develop mechanisms for ensuring the stability of the RNNs with controlled elements and to evaluate their effectiveness in the implementation of neural network forecasting.

2. Architecture of the software system for neural network forecasting
The architecture of the software system that implements the neural network time series forecasting method is shown in Figure 1. Its peculiarity is that it consists of two identical two-layer neural networks with controlled elements. According to the method, time series are converted into sequences of sets of single patterns (SSPs), which is the responsibility of the preprocessing module.

![Fig. 1. Architecture of the software system for neural network forecasting.](image)

Then the SSPs enter the first RNN and shift along the layers from the input to the output of neural network, as a result of which a space-time model is formed at the synapses of the first RNN. When it is necessary to get a forecast, the state of the first RNN is copied to the second RNN by the command from the control unit. While the first RNN continues training, the second RNN starts processing the copied SSPs in parallel mode. The internal time of the second RNN repeatedly accelerates, and the associative call of signals in the direction of the input strengthens. At the same time, new SSPs are formed on the layers of the second RNN that carry forecast values. After removing SSPs from the output of the second RNN, they are decoded in the post-processing module. The control unit functions are implemented in the parameter read / write modules, button panels, and modules for RNN layer states visualizing.

3. The stability control mechanisms

3.1. The stability criterion
By the stability of the neural network time series forecasting system, we will understand the state of its constituent RNNs with controlled elements, in which the total weight of $w_{\text{sum}}(t)$ of all its synapses is within the limits $[-W, +W]$, where $W$ is a certain positive coefficient:

$$w_{\text{sum}}(t) = \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij}(t) \epsilon [-W, +W].$$

Here, the weights of the $w_{ij}(t)$ synapses depend on the weight coefficients $k_{ij}(t)$, which determine the conductivity of the RNN synapses. In turn, $k_{ij}(t)$ is defined by the equation:

$$k_{ij}(t) = \frac{(1+\exp(-\gamma g_{ij}(t)))/(1+\exp(-\gamma g_{ij}(t))) = \text{th}(\gamma g_{ij}(t)/2),}{(2)}$$

$$g_{ij}(t) = g_{ij}(t-\Delta t) + \Delta g_{ij}(t).$$

$$(3)$$
Here \( g_i(t) \) and \( g_i(t-\Delta t) \) are the prehistories at time points \( t \) and \( t-\Delta t \), respectively; \( g_i(0)=0; \gamma \) is some positive coefficient. The value of \( \Delta g_i(t) \) is determined depending on the states of the \( i \)-th and \( j \)-th neurons. If the \( i \)-th neuron emitted a signal and then the \( j \)-th neuron was excited, then \( \Delta g_{ij}(t) \) is a positive value \( (g') \). If the \( j \)-th neuron was excited without a signal from the \( i \)-th neuron, then \( \Delta g_{ij}(t) \) is negative \( (g') \). In all other cases, \( \Delta g_{ij}(t)=0 \).

3.2. The stability criterion

It can be seen from equations (1) – (3) that the stability of the RNN depends primarily on the values of \( g_i(t) \), on the basis of which the weight coefficients are calculated. At each step of the RNN functioning, according to the learning rule, there are a certain number of synapses for which \( \Delta g_{ij}(t) = g' \), and synapses for which \( \Delta g_{ij}(t) = g' \). The first condition for stability is to maintain a balance of the total learning history. To do this, we introduce the balance equation:

\[
S'(t) \cdot g'(t) - S(t) \cdot g(t) = V. \tag{4}
\]

Here \( V \) is a constant value (for example is 0); \( S'(t) \) and \( S(t) \) are the numbers of synapses for which \( \Delta g_{ij}(t) = g'(t) \) and \( \Delta g_{ij}(t) = g(t) \), respectively.

Further, synapses with weights less than 0 are conventionally called "inhibitory", and synapses with a positive weight are called "excitatory".

3.3. The modified learning rule for the neural network

The dependence of \( k_i(t) \) on \( g_i(t) \) is nonlinear, so the balance of the total learning history is insufficient to ensure the stability of the RNN. An additional requirement is the balance of the "excitatory" and "inhibitory" potential of synapses. In other words, at each step of the RNN training, it is necessary that the number of synapses whose weight has decreased in accordance with the training rule be equal to the number of synapses with increased weight. At the same time, the current learning rule does not meet this requirement, since the number of "excitatory" and "inhibitory" synapses depends on the ratio of excited and waiting neurons, which may change over time. One of the ways to equalize the number of "excitatory" and "inhibitory" synapses is to modify the learning rule. In comparison with the original rule, it is proposed to reduce the weight of the \( ij \)-th synapse if the \( i \)-th neuron was in a state of refractoriness, and at the next moment the \( j \)-th neuron was excited. In this case, the number of refractive cycles of neurons is selected so that the number of excited neurons in the layer is equal to the number of active ones.

4. Experiments

The proposed solutions were checked on the example of forecasting urban road traffic in St. Petersburg. To do this, the RNN was configured with the size of 5×6 logical fields, each of which had the size of 4 × 8 neurons (Figure 2). The depth of forecasts was 12 days, and the horizon was 2 days at an interval of 3 hours.

The first series of experiments was conducted without using stability control mechanisms. Figure 3 shows the dynamics of the total weight of the RNN. The horizontal axis of the graphs shows the RNN operation cycles and the vertical axis shows the values of the corresponding parameters. The graphs show that the total weight coefficient goes to the negative range of values. This is due to the prevalence of the number of "inhibitory" synapses over "excitatory" ones (in Figure 2 on the average only 4 out of 32 neurons are excited in one logical field).

The second series was conducted using stability control mechanisms. Figure 4 shows the graph for this situation. It can be seen that the total weight is at a constant level, and the fluctuations in the total weight do not go beyond -60...+60 (against -5000...0 in the previous series).

According to the results of experiments, stability control mechanisms allowed reducing the error of MAE by 13.8%, MAPE by 10.5% and RMSE by 10.0%, which indicates the effectiveness of the proposed solutions.
Fig. 2. State of the first layer of the second RNN when predicting traffic flows. The white serpentine arrow shows the route of data movement from the input to the output of the RNN. White circles indicate neurons in the waiting state, black circles indicate active neurons, and gray circles indicate refractor neurons.

Fig. 3. Dynamics of the total weight coefficient in RNN training without stability control mechanisms.

Fig. 4. Dynamics of the total weight coefficient in RNN training with stability control mechanisms.

5. Conclusions
The considered RNN with controlled elements has a more flexible structure due to the expansion of signal routing capabilities and element relationships. Increasing the manageability of the network makes it necessary to monitor its stability in the learning process, for which the stability criterion and
control mechanisms have been developed. Numerical examples showing the total weight of the neural network during training demonstrate that the proposed mechanisms meet the necessary requirements.

References
[1] Hyndman R J and Athanasopoulos G 2018 Forecasting: principles and practice 2nd ed Melbourne, Australia
[2] Polson N G and Sokolov V O 2017 Deep learning for short-term traffic flow prediction Transportation research Part C 79 p 1–17. http://dx.doi.org/10.1016/j.trc.2017.02.024
[3] Fernandez-Navarro M, Cruz M A, Gutierrez P A, Castano A and Hervas-Martinez C 2018 Time series forecasting by recurrent product unit neural networks Neural Comput & Applic 29 p 779–791 https://doi.org/10.1007/s00521-016-2494-2
[4] Goudarzi S, Kama M N, Anisi M H, Soleymani S A and Doctor F 2018 Self-organizing traffic flow prediction with an optimized deep belief network for Internet of vehicles Sensors 18 p 3459 doi:10.3390/s18103459
[5] Osipov V and Osipova M 2018 Space–time signal binding in recurrent neural networks with controlled elements Neurocomputing 308 p 194–204 https://doi.org/10.1016/j.neucom.2018.05.009
[6] Wang H, Wei G, Wen S and Huang T 2020 Generalized norm for existence, uniqueness and stability of Hopfield neural networks with discrete and distributed delays Comparative Study Neural Netw p 667–675
[7] Yin H, Seiler P and Arcak M 2020 Stability Analysis using Quadratic Constraints for Systems with Neural Network Controllers Electrical Engineering and Systems Science
[8] Wang F, Liu X, Tang M and Chen L 2019 Further results on stability and synchronization of fractional-order Hopfield neural networks Neurocomputing 346 p 12–19 https://doi.org/10.1016/j.neucom.2018.08.089
[9] Cui N, Jiang H, Hu C and Abdurahman A 2018 Global asymptotic and robust stability of inertial neural networks with proportional delays Neurocomputing 272 p 326–333 https://doi.org/10.1016/j.neucom.2017.07.001
[10] Lu S and Gao Y 2019 Exponential stability in Lagrange sense for inertial neural networks with time-varying delays Neurocomputing 333 p 41–52 https://doi.org/10.1016/j.neucom.2018.12.063
[11] Liu P and Cao G 2020 Employing the Friedrichs’ inequality to ensure global exponential stability of delayed reaction-diffusion neural networks with nonlinear boundary conditions Neurocomputing 383 p 81–94 https://doi.org/10.1016/j.neucom.2019.11.091
[12] Osipov V and Miloserdov D 2020 Neural network forecasting of traffic congestion Digital Transformation and Global Society Springer Cham vol 1038 https://doi.org/10.1007/978-3-030-37858-5_20
[13] Osipov V, Zhukova N and Miloserdov D 2019 Neural Network Associative Forecasting of Demand for Goods Experimental Economics and Machine Learning Perm vol 2479