Method for Recognizing Linguistic Constructions Based on Stochastic Neural Networks

Eugene Fedorov and Olga Nechyporenko

Cherkasy State Technological University, Shevchenko blvd., 460, Cherkasy, 18006, Ukraine

Abstract

The paper proposes a method for recognizing linguistic constructions based on stochastic neural networks. The novelty of the study lies in the fact that in order to ensure the interaction of software agents representing subjects that operate within supply chains, two models of an artificial neural network were created to recognize natural language structures based on the restricted Boltzmann machine (in contrast to it, the neurons of the hidden layer were interconnected), a criterion for evaluating the effectiveness of training the proposed models was chosen, the parameters of the proposed models were identified based on the contrastive divergence. The proposed models and methods for their parametric identification make it possible to improve the recognition accuracy of natural language constructions. The proposed method for recognizing linguistic structures based on stochastic neural networks can be used in various intelligent systems that use the recognition of natural language structures.

Keywords

supply chain, multi-agent interaction, artificial neural network, restricted Boltzmann machine, contrastive divergence, linguistic constructions

1. Introduction

Currently, one of the most urgent problems in the field of processing natural language structures is insufficiently high speed, adequacy and probability of correct recognition [1, 2]. This leads to the fact that the interaction of software agents in multi-agent natural language systems may be inefficient. Therefore, the development of methods that increase the efficiency of using linguistic structures for the interaction of software agents is an urgent task.

In this work, the interaction of software agents representing subjects that operate within supply chains is chosen as the scope of natural language constructions [3-5].

To date, there are no supply chain management computer systems that provide modeling of the interaction of logistics entities based on soft calculations and linguistic structures.

Currently, artificial intelligence methods are used to recognize natural language constructions, with the most popular being artificial neural networks [6-8].

The advantages of neural networks are [9-11]:

- the possibility of their training and adaptation;
- the ability to identify patterns in the data, their generalization, i.e. knowledge extraction from data, hence no knowledge about the object is required (for example, its mathematical model);
- parallel processing of information that increases computing power.

The disadvantages of neural networks are [12-14]:

- difficulty in determining the structure of the network, since there are no algorithms for calculating the number of layers and neurons in each layer for specific applications;

COLINS-2022: 6th International Conference on Computational Linguistics and Intelligent Systems, May 12–13, 2022, Gliwice, Poland
EMAIL: fedorovee75@ukr.net (E. Fedorov); olne@ukr.net (O. Nechyporenko)
ORCID: 0000-0003-3841-7373 (E. Fedorov); 0000-0002-3954-3796 (O. Nechyporenko)

- difficulty in forming a representative sample;
- high probability for the training and adaptation method hitting a local extremum;
- inaccessibility to human understanding of the knowledge accumulated by the network (it is impossible to represent the relationship between input and output in the form of rules), since they are distributed among all elements of the neural network and are presented in the form of its weight coefficients.

The following recurrent networks are most often used as neural networks for recognition:

- Elman neural network (ENN) or simple recurrent network (SRN) [15, 16], which is a recurrent two-layer network and is based on a multilayer perceptron (MLP). The advantages of this network are a simpler architecture and higher learning rate than in gated and bidirectional networks. The disadvantage is the insufficient recognition accuracy compared to bidirectional networks;
- bidirectional recurrent neural network (BRNN) [17, 18], which is a recurrent two-layer network and is built on the basis of two Elman neural networks. The advantage of this network is a higher recognition accuracy than in a conventional Elman neural network. The disadvantages are a higher complexity of determining the architecture, a lower learning rate than in a conventional Elman neural network;
- long short-term memory (LSTM) [19, 20], which is a recurrent network and is built on the basis of memory blocks (containing one or more cells) and input, output forgetting gates (FIR filters). The advantage of this network is a higher recognition accuracy than in a conventional Elman neural network. The disadvantages are a higher complexity of determining the architecture, a lower learning rate than in a conventional Elman neural network;
- bidirectional recurrent neural network (BLSTM) [21, 22], which is a recurrent network and is built on the basis of two LSTM neural networks. The advantage of this network is a higher recognition accuracy than in a conventional LSTM. The disadvantages are a higher architecture definition complexity, a lower learning rate than conventional LSTM;
- gated recurrent unit (GRU) [23, 24], which is a recurrent two-layer network and is built on the basis of hidden blocks and reset and update gates (FIR filters). The advantage of this network is a higher recognition accuracy than in a conventional Elman neural network. The disadvantages are a higher complexity of determining the architecture, a lower learning rate than in a conventional Elman neural network;
- idirectional recurrent neural network (BGRU) [25], which is a recurrent network and is built on the basis of two GRU neural networks. The advantage of this network is a higher recognition accuracy than in a conventional GRU. The disadvantages are a higher complexity of architecture definition, a lower learning rate than in conventional GRU.

Thus, none of the networks satisfies all the criteria.

The aim of the work is to develop a method for recognizing natural language constructions. To achieve the goal, the following tasks were set and solved:

- analyze existing recognition methods;
- propose neural network recognition models;
- choose a criterion for evaluating the effectiveness of neural network recognition models;
- propose methods for determining the values of the parameters of neural network recognition models;
- conduct numerical study.

2. Block diagram of neural network recognition models

Figure 1-2 shows block diagrams of recognition models based on the unidirectional and bidirectional restricted Boltzmann machine with a recurrent hidden layer (RBMRHL), which is a recurrent ANN and contains one visible layer and one hidden layer. Unlike the traditional restricted Boltzmann machine (RBM) [26–27], the hidden layer is recurrent. This makes it possible to improve recognition accuracy.

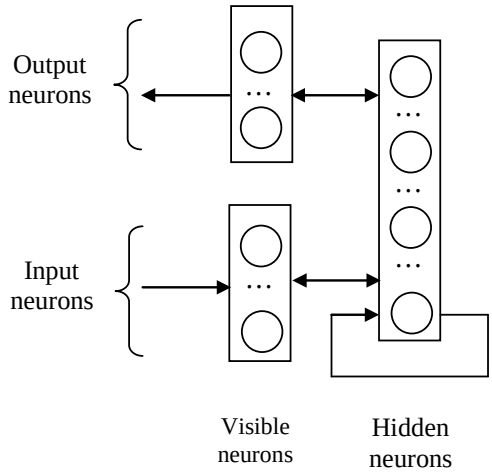

Figure 1: Block diagram of a recognition model based on a unidirectional restricted machine with a recurrent hidden layer (RBMRHL)

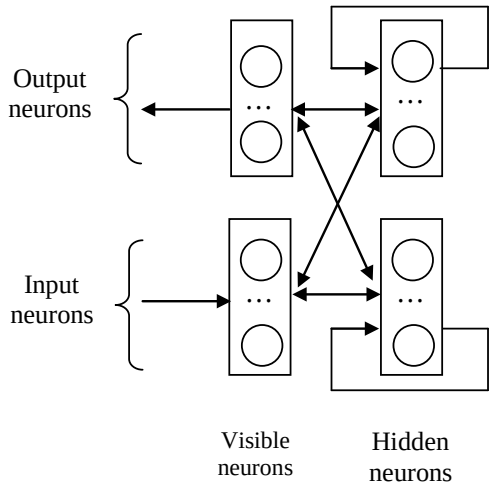

Figure 2: Block diagram of a recognition model based on a bidirectional restricted machine with a recurrent hidden layer (RBMRHL)

3. Neural network recognition models

3.1. Recognition model based on a unidirectional RBMRHL

Positive phase (steps 1-2)

1. $\mathbf{x}^{in} = \mathbf{x}_1^{in}$, $\mathbf{x}^{out} = \mathbf{0}$.
2. Computation of the state of hidden neurons

$$P_j = \frac{1}{1 + \exp\left(- b_j^h - \sum_{i=1}^{N^{in}} w_{ij}^{in-h} x_i^{in} - \sum_{i=1}^{N^{out}} w_{ij}^{out-h} x_i^{out} - \sum_{i=1}^{N^h} w_{ij}^{h-h} x_i^h \right)}, \quad j \in \overline{1, N^h},$$

$$x_j^h = \begin{cases} 1, & P_j \geq U(0,1) \\ 0, & P_j < U(0,1) \end{cases}, \quad j \in \overline{1, N^h}.$$

Negative phase (step 3)

3. Computation of the state of visible output neurons

$$P_j = \frac{1}{1+\exp\left(-b_j^{out} - \sum_{i=1}^{N^h} w_{ij}^{out-h} x_i^h\right)},\ j \in \overline{1, N^{out}},$$

$$x_j^{out} = \begin{cases} 1, & P_j \geq U(0,1) \\ 0, & P_j < U(0,1) \end{cases},\ j \in \overline{1, N^{out}}.$$

The result is vector $(x_1^{out}, ..., x_{N^{out}}^{out})$.

3.2. Recognition model based on a bidirectional RBMRHL

Positive phase (steps 1-2)

1. $\mathbf{x}^{in} = \mathbf{x}_1^{in}$, $\mathbf{x}^{out} = \mathbf{0}$
2. Computation of the state of hidden neurons

$$P_j = \frac{1}{1+\exp\left(-b_j^h - \sum_{i=1}^{N^{in}} wL_{ij}^{in-h} x_i^{in} - \sum_{i=1}^{N^{out}} wL_{ij}^{out-h} x_i^{out} - \sum_{i=1}^{N^h} wL_{ij}^{h-h} xL_i^h\right)},\ j \in \overline{1, N^h},$$

$$xL_j^h = \begin{cases} 1, & P_j \geq U(0,1) \\ 0, & P_j < U(0,1) \end{cases},\ j \in \overline{1, N^h},$$

$$P_j = \frac{1}{1+\exp\left(-b_j^h - \sum_{i=1}^{N^{in}} wR_{ij}^{in-h} x_i^{in} - \sum_{i=1}^{N^{out}} wR_{ij}^{out-h} x_i^{out} - \sum_{i=1}^{N^h} wR_{ij}^{h-h} xR_i^h\right)},\ j \in \overline{1, N^h},$$

$$xR_j^h = \begin{cases} 1, & P_j \geq U(0,1) \\ 0, & P_j < U(0,1) \end{cases},\ j \in \overline{1, N^h}.$$

Negative phase (step 3)

3. Computation of the state of visible output neurons

$$P_j = \frac{1}{1+\exp\left(-b_j^{out} - \sum_{i=1}^{N^h} wL_{ij}^{out-h} xL_i^h - \sum_{i=1}^{N^h} wR_{ij}^{out-h} xR_i^h\right)},\ j \in \overline{1, N^{out}},$$

$$x_j^{out} = \begin{cases} 1, & P_j \geq U(0,1) \\ 0, & P_j < U(0,1) \end{cases},\ j \in \overline{1, N^{out}}.$$

The result is vector $(x_1^{out}, ..., x_{N^{out}}^{out})$.

4. Criteria for evaluating the effectiveness of neural network recognition models

In this work, for training a unidirectional and bidirectional RBMRHL model, the model adequacy criterion was chosen, which means choosing such parameter values $W = \{w_{ij}^{in-h}, w_{ij}^{out-h}, w_{ij}^{h-h}\}$ or $W = \{wL_{ij}^{in-h}, wR_{ij}^{in-h},\ wL_{ij}^{out-h}, wR_{ij}^{out-h},\ wL_{ij}^{h-h}, wR_{ij}^{h-h}\}$ respectively, that deliver maximum accuracy (matching the model output and the desired output):

$$F = \frac{1}{P}\sum_{\mu=1}^{P}\left[\mathbf{d}_\mu = \mathbf{x}_\mu^{out}\right] \to \max_W,\ \left[\mathbf{d}_\mu = \mathbf{y}_\mu\right] = \begin{cases} 1, & \mathbf{d}_\mu = \mathbf{y}_\mu \\ 0, & \mathbf{d}_\mu \neq \mathbf{y}_\mu \end{cases} \quad (1)$$

The training of the RBMRHL model is subject to the criterion (1).

5. Methods for determining the values of parameters of neural network recognition models

5.1. Principles for determining the parameters of neural network recognition models

RBMRHL parameter values are determined based on the CD-1 contrastive divergence method, which speeds up supervised learning, since instead of stabilizing the state of neurons, it performs only one step of tuning their state. RBMRHL classification operates in two phases - positive and negative.

For RBMRHL recognition in the positive phase, visible input and output neurons are fixed, and RBMRHL functions until hidden neurons are established. In the negative phase, firstly hidden neurons trained in the positive phase are fixed, and RBMRHL functions until visible input and output neurons are established, after which visible input and output neurons trained in the negative phase are fixed, and RBMRHL functions until hidden neurons are established.

5.2. Method for determining the parameter values of a unidirectional RBMRHL model for recognition based on contrastive divergence

1. Number of training iteration $n=1$, initialization by means of uniform distribution on the interval (0,1) or [-0.5, 0.5] offsets (thresholds) $b_i^{in}(n)$, $i \in \overline{1, N^{in}}$, $b_i^{out}(n)$, $i \in \overline{1, N^{out}}$, $b_j^h(n)$, $j \in \overline{1, N^h}$, and weights $w_{ij}^{in-h}(n)$, $i \in \overline{1, N^{in}}$, $j \in \overline{1, N^h}$, $w_{ij}^{out-h}(n)$, $i \in \overline{1, N^{out}}$, $j \in \overline{1, N^h}$, $w_{ij}^{h-h}(n)$, $i \in \overline{1, N^h}$, $j \in \overline{1, N^h}$, $w_{ii}^{in-h}(n)=0$, $w_{ii}^{out-h}(n)=0$, $w_{ij}^{h-h}(n)=0$, $w_{ij}^{in-h}(n)=w_{ji}^{in-h}(n)$, $w_{ij}^{out-h}(n)=w_{ji}^{out-h}(n)$, $w_{ij}^{h-h}(n)=w_{ji}^{h-h}(n)$.
2. Training set $\{(\mathbf{x}_\mu^{in}, \mathbf{x}_\mu^{out}) \mid \mathbf{x}_\mu^{in} \in \{0,1\}^{N^{in}}, \mathbf{x}_\mu^{out} \in \{0,1\}^{N^{out}}\}$, $\mu \in \overline{1, P}$ is specified, where $\mathbf{x}_\mu^{in}$ – μ^{th} training vector of states of visible input neurons, $\mathbf{x}_\mu^{out}$ – μ^{th} training vector of states of visible output neurons, P is the power of the training set.

Positive phase (steps 3-7)

3. $\mathbf{x}1_\mu^{in} = \mathbf{x}_\mu^{in}$, $\mathbf{x}1_\mu^{out} = \mathbf{x}_\mu^{out}$, $\mu \in \overline{1, P}$.
4. $\mu=1$, $\mathbf{x}1_{\mu-1}^h = \mathbf{0}$,
5. $\mathbf{x}^{in} = \mathbf{x}1_\mu^{in}$, $\mathbf{x}^{out} = \mathbf{x}1_\mu^{out}$, $\mathbf{x}^h = \mathbf{x}1_{\mu-1}^h$.
6. Computation of the state of hidden neurons

$$P_j = \frac{1}{1+\exp\left(\begin{array}{l} -b_j^h(n) - \sum_{i=1}^{N^{in}} w_{ij}^{in-h}(n) x_i^{in} - \\ \sum_{i=1}^{N^{out}} w_{ij}^{out-h}(n) x_i^{out} - \sum_{i=1}^{N^h} w_{ij}^{h-h}(n) x_i^h \end{array}\right)}, \quad j \in \overline{1, N^h},$$

$$x_j^h = \begin{cases} 1, & P_j \geq U(0,1) \\ 0, & P_j < U(0,1) \end{cases}, \quad j \in \overline{1, N^h}.$$

7. $\mathbf{x}1_\mu^h = \mathbf{x}^h$. If $\mu < P$, then $\mu = \mu + 1$, go to 5.

Negative phase (steps 8-16)

8. $\mu=1$.
9. $\mathbf{x}^h = \mathbf{x}1_\mu^h$.
10. Computation of the state of visible output neurons

$$P_j = \frac{1}{1+\exp\left(-b_j^{in}(n) - \sum_{i=1}^{N^h} w_{ij}^{in-h}(n) x_i^h\right)},\ j \in \overline{1, N^{in}},$$

$$x_j^{in} = \begin{cases} 1, & P_j \geq U(0,1) \\ 0, & P_j < U(0,1) \end{cases},\ j \in \overline{1, N^{in}}.$$

11. Computation of the state of visible input neurons

$$P_j = \frac{1}{1+\exp\left(-b_j^{out}(n) - \sum_{i=1}^{N^h} w_{ij}^{out-h}(n) x_i^h\right)},\ j \in \overline{1, N^{out}},$$

$$x_j^{out} = \begin{cases} 1, & P_j \geq U(0,1) \\ 0, & P_j < U(0,1) \end{cases},\ j \in \overline{1, N^{out}}.$$

12. $\mathbf{x}2_{\mu}^{in} = \mathbf{x}^{in}$, $\mathbf{x}2_{\mu}^{out} = \mathbf{x}^{out}$. If $\mu < P$, then $\mu = \mu + 1$, go to 9.

13. $\mu = 1$, $\mathbf{x}2_{\mu-1}^{h} = \mathbf{0}$,

14. $\mathbf{x}^{in} = \mathbf{x}2_{\mu}^{in}$, $\mathbf{x}^{out} = \mathbf{x}2_{\mu}^{out}$, $\mathbf{x}^{h} = \mathbf{x}2_{\mu-1}^{h}$.

15. Computation of the state of hidden neurons

$$P_j = \frac{1}{1+\exp\left(\begin{array}{l} -b_j^h(n) - \sum_{i=1}^{N^{in}} w_{ij}^{in-h}(n) x_i^{in} - \\ \sum_{i=1}^{N^{out}} w_{ij}^{out-h}(n) x_i^{out} - \sum_{i=1}^{N^h} w_{ij}^{h-h}(n) x_i^h \end{array}\right)},\ j \in \overline{1, N^h},$$

$$x_j^h = \begin{cases} 1, & P_j \geq U(0,1) \\ 0, & P_j < U(0,1) \end{cases},\ j \in \overline{1, N^h}.$$

16. $\mathbf{x}2_{\mu}^{h} = \mathbf{x}^{h}$. If $\mu < P$, then $\mu = \mu + 1$, go to 14.

17. Adjustment of synaptic weights based on Boltzmann's rule

$$b_i^{in}(n+1) = b_i^{in}(n) + \eta\left(\frac{1}{P}\sum_{\mu=1}^{P} x1_{\mu i}^{in} - \frac{1}{P}\sum_{\mu=1}^{P} x2_{\mu i}^{in}\right),\ i \in \overline{1, N^{in}},$$

$$b_i^{out}(n+1) = b_i^{out}(n) + \eta\left(\frac{1}{P}\sum_{\mu=1}^{P} x1_{\mu i}^{out} - \frac{1}{P}\sum_{\mu=1}^{P} x2_{\mu i}^{out}\right),\ i \in \overline{1, N^{out}},$$

$$b_j^{h}(n+1) = b_j^{h}(n) + \eta\left(\frac{1}{P}\sum_{\mu=1}^{P} x1_{\mu j}^{h} - \frac{1}{P}\sum_{\mu=1}^{P} x2_{\mu j}^{h}\right),\ j \in \overline{1, N^{h}},$$

$$w_{ij}^{in-h}(n+1) = w_{ij}^{in-h}(n) + \eta(\rho_{ij}^{+} - \rho_{ij}^{-}),\ i \in \overline{1, N^{in}},\ j \in \overline{1, N^{h}},$$

$$\rho_{ij}^{+} = \frac{1}{P}\sum_{\mu=1}^{P} x1_{\mu i}^{in} x1_{\mu j}^{h},\ \rho_{ij}^{-} = \frac{1}{P}\sum_{\mu=1}^{P} x2_{\mu i}^{in} x2_{\mu j}^{h},$$

$$w_{ij}^{out-h}(n+1) = w_{ij}^{out-h}(n) + \eta(\rho_{ij}^{+} - \rho_{ij}^{-}),\ i \in \overline{1, N^{out}},\ j \in \overline{1, N^{h}},$$

$$\rho_{ij}^{+} = \frac{1}{P}\sum_{\mu=1}^{P} x1_{\mu i}^{out} x1_{\mu j}^{h},\ \rho_{ij}^{-} = \frac{1}{P}\sum_{\mu=1}^{P} x2_{\mu i}^{out} x2_{\mu j}^{h},$$

$$w_{ij}^{h-h}(n+1) = w_{ij}^{h-h}(n) + \eta(\rho_{ij}^{+} - \rho_{ij}^{-}),\ i \in \overline{1, N^{h}},\ j \in \overline{1, N^{h}},$$

$$\rho_{ij}^{+}=\frac{1}{P}\sum_{\mu=1}^{P}x1_{\mu-1,i}^{h}x1_{\mu j}^{h},\ \rho_{ij}^{-}=\frac{1}{P}\sum_{\mu=1}^{P}x2_{\mu-1,i}^{h}x2_{\mu j}^{h}.$$

18. If $\frac{1}{P\cdot N^{out}}\sum_{\mu=1}^{P}\sum_{i=1}^{N^{out}}|\,x1_{\mu i}^{out}-x2_{\mu i}^{out}\,|>\varepsilon$, then $n=n+1$, go to 2.

5.3. Method for determining parameter values of a bidirectional RBMRHL model for recognition based on contrastive divergence

1. Number of training iteration $n=1$, initialization by means of uniform distribution on the interval (0,1) or [-0.5, 0.5] offsets (thresholds) $b_i^{in}(n)$, $i\in\overline{1,N^{in}}$, $b_i^{out}(n)$, $i\in\overline{1,N^{out}}$, $bL_j^h(n)$, $j\in\overline{1,N^h}$, $bR_j^h(n)$, $j\in\overline{1,N^h}$, and weights $wL_{ij}^{in-h}(n)$, $i\in\overline{1,N^{in}}$, $j\in\overline{1,N^h}$, $wR_{ij}^{in-h}(n)$, $i\in\overline{1,N^{in}}$, $j\in\overline{1,N^h}$, $wL_{ij}^{out-h}(n)$, $i\in\overline{1,N^{out}}$, $j\in\overline{1,N^h}$, $wR_{ij}^{out-h}(n)$, $i\in\overline{1,N^{out}}$, $j\in\overline{1,N^h}$, $wL_{ij}^{h-h}(n)$, $i\in\overline{1,N^h}$, $j\in\overline{1,N^h}$, $wR_{ij}^{h-h}(n)$, $i\in\overline{1,N^h}$, $j\in\overline{1,N^h}$, $wL_{ii}^{in-h}(n)=0$, $wR_{ii}^{in-h}(n)=0$, $wL_{ii}^{out-h}(n)=0$, $wR_{ii}^{out-h}(n)=0$, $wL_{ij}^{in-h}(n)=wL_{ji}^{in-h}(n)$, $wR_{ij}^{in-h}(n)=wR_{ji}^{in-h}(n)$, $wL_{ij}^{out-h}(n)=wL_{ji}^{out-h}(n)$, $wR_{ij}^{out-h}(n)=wR_{ji}^{out-h}(n)$.

2. Training set $\{(\mathbf{x}_\mu^{in},\mathbf{x}_\mu^{out})\,|\,\mathbf{x}_\mu^{in}\in\{0,1\}^{N^{in}},\mathbf{x}_\mu^{out}\in\{0,1\}^{N^{out}}\}$, $\mu\in\overline{1,P}$, is specified, where $\mathbf{x}_\mu^{in}$ – μ^{th} training vector of states of visible input neurons, $\mathbf{x}_\mu^{out}$ – μ^{th} training vector of states of visible output neurons, P is the power of the training set.

Positive phase (steps 3-11)

3. $\mathbf{x}1_\mu^{in}=\mathbf{x}_\mu^{in}$, $\mathbf{x}1_\mu^{out}=\mathbf{x}_\mu^{out}$, $\mu\in\overline{1,P}$.
4. $\mu=1$, $\mathbf{xL}1_{\mu-1}^{h}=\mathbf{0}$,
5. $\mathbf{x}^{in}=\mathbf{x}1_\mu^{in}$, $\mathbf{x}^{out}=\mathbf{x}1_\mu^{out}$, $\mathbf{xL}^{h}=\mathbf{xL}1_{\mu-1}^{h}$
6. Computation of the state of hidden neurons

$$P_j=\frac{1}{1+\exp\left(\begin{array}{l}-bL_j^h(n)-\sum_{i=1}^{N^{in}}wL_{ij}^{in-h}(n)x_i^{in}-\sum_{i=1}^{N^{out}}wL_{ij}^{out-h}(n)x_i^{out}-\\ \sum_{i=1}^{N^h}wL_{ij}^{h-h}(n)xL_i^h\end{array}\right)},\ j\in\overline{1,N^h},$$

$$xL_j^h=\begin{cases}1, & P_j\ge U(0,1)\\ 0, & P_j<U(0,1)\end{cases},\ j\in\overline{1,N^h}.$$

7. $\mathbf{xL}1_\mu^{h}=\mathbf{xL}^{h}$. If $\mu<P$, then $\mu=\mu+1$, go to 5.
8. $\mu=P$, $\mathbf{xR}1_{\mu+1}^{h}=\mathbf{0}$.
9. $\mathbf{x}^{in}=\mathbf{x}1_\mu^{in}$, $\mathbf{x}^{out}=\mathbf{x}1_\mu^{out}$, $\mathbf{xR}^{h}=\mathbf{xR}1_{\mu+1}^{h}$.
10. Computation of the state of hidden neurons

$$P_j = \frac{1}{1+\exp\left(\begin{array}{l} -bR_j^h(n) - \sum_{i=1}^{N^{in}} wR_{ij}^{in-h}(n)x_i^{in} - \sum_{i=1}^{N^{out}} wR_{ij}^{out-h}(n)x_i^{out} - \\ \sum_{i=1}^{N^h} wR_{ij}^{h-h}(n)xR_i^h \end{array}\right)}, \quad j \in \overline{1, N^h},$$

$$xR_j^h = \begin{cases} 1, & P_j \geq U(0,1) \\ 0, & P_j < U(0,1) \end{cases}, \quad j \in \overline{1, N^h}.$$

11. $\mathbf{xR}1_\mu^h = \mathbf{xR}^h$. If $\mu > 1$, then $\mu = \mu - 1$, go to 9.

Negative phase (steps 12-24)

12. $\mu = 1$.
13. $\mathbf{xL}^h = \mathbf{xL}1_\mu^h$, $\mathbf{xR}^h = \mathbf{xR}1_\mu^h$.
14. Computation of the state of visible output neurons

$$P_j = \frac{1}{1+\exp\left(-b_j^{in}(n) - \sum_{i=1}^{N^h} wL_{ij}^{in-h}(n)xL_i^h - \sum_{i=1}^{N^h} wR_{ij}^{in-h}(n)xR_i^h\right)}, \quad j \in \overline{1, N^{in}},$$

$$x_j^{in} = \begin{cases} 1, & P_j \geq U(0,1) \\ 0, & P_j < U(0,1) \end{cases}, \quad j \in \overline{1, N^{in}}.$$

15. Computation of the state of visible output neurons

$$P_j = \frac{1}{1+\exp\left(-b_j^{out}(n) - \sum_{i=1}^{N^h} wL_{ij}^{out-h}(n)xL_i^h - \sum_{i=1}^{N^h} wR_{ij}^{out-h}(n)xR_i^h\right)}, \quad j \in \overline{1, N^{out}},$$

$$x_j^{out} = \begin{cases} 1, & P_j \geq U(0,1) \\ 0, & P_j < U(0,1) \end{cases}, \quad j \in \overline{1, N^{out}}.$$

16. $\mathbf{x}2_\mu^{in} = \mathbf{x}^{in}$, $\mathbf{x}2_\mu^{out} = \mathbf{x}^{out}$. If $\mu < P$, then $\mu = \mu + 1$, go to 13.
17. $\mu = 1$, $\mathbf{xL}2_{\mu-1}^h = \mathbf{0}$,
18. $\mathbf{x}^{in} = \mathbf{x}2_\mu^{in}$, $\mathbf{x}^{out} = \mathbf{x}2_\mu^{out}$, $\mathbf{xL}^h = \mathbf{xL}2_{\mu-1}^h$
19. Computation of the state of hidden neurons

$$P_j = \frac{1}{1+\exp\left(\begin{array}{l} -bL_j^h(n) - \sum_{i=1}^{N^{in}} wL_{ij}^{in-h}(n)x_i^{in} - \sum_{i=1}^{N^{out}} wL_{ij}^{out-h}(n)x_i^{out} - \\ \sum_{i=1}^{N^h} wL_{ij}^{h-h}(n)xL_i^h \end{array}\right)}, \quad j \in \overline{1, N^h},$$

$$xL_j^h = \begin{cases} 1, & P_j \geq U(0,1) \\ 0, & P_j < U(0,1) \end{cases}, \quad j \in \overline{1, N^h}.$$

20. $\mathbf{xL}2_\mu^h = \mathbf{xL}^h$. If $\mu < P$, then $\mu = \mu + 1$, go to 18.
21. $\mu = P$, $\mathbf{xR}1_{\mu+1}^h = \mathbf{0}$.
22. $\mathbf{x}^{in} = \mathbf{x}2_\mu^{in}$, $\mathbf{x}^{out} = \mathbf{x}2_\mu^{out}$, $\mathbf{xR}^h = \mathbf{xR}2_{\mu+1}^h$.
23. Computation of the state of hidden neurons

$$P_j = \frac{1}{1+\exp\left(\begin{array}{l} -bR_j^h(n) - \sum_{i=1}^{N^{in}} wR_{ij}^{in-h}(n)x_i^{in} - \sum_{i=1}^{N^{out}} wR_{ij}^{out-h}(n)x_i^{out} - \\ \sum_{i=1}^{N^h} wR_{ij}^{h-h}(n)xR_i^h \end{array}\right)}, \ j \in \overline{1, N^h},$$

$$xR_j^h = \begin{cases} 1, & P_j \geq U(0,1) \\ 0, & P_j < U(0,1) \end{cases}, \ j \in \overline{1, N^h}.$$

24. $\mathbf{xR}2_\mu^h = \mathbf{xR}^h$. If $\mu > 1$, then $\mu = \mu - 1$, go to 22.

25. Adjustment of synaptic weights based on Boltzmann's rule

$$b_i^{in}(n+1) = b_i^{in}(n) + \eta\left(\frac{1}{P}\sum_{\mu=1}^{P} x1_{\mu i}^{in} - \frac{1}{P}\sum_{\mu=1}^{P} x2_{\mu i}^{in}\right), \ i \in \overline{1, N^{in}},$$

$$b_i^{out}(n+1) = b_i^{out}(n) + \eta\left(\frac{1}{P}\sum_{\mu=1}^{P} x1_{\mu i}^{out} - \frac{1}{P}\sum_{\mu=1}^{P} x2_{\mu i}^{out}\right), \ i \in \overline{1, N^{out}},$$

$$bL_j^h(n+1) = bL_j^h(n) + \eta\left(\frac{1}{P}\sum_{\mu=1}^{P} xL1_{\mu j}^h - \frac{1}{P}\sum_{\mu=1}^{P} xL2_{\mu j}^h\right), \ j \in \overline{1, N^h},$$

$$bR_j^h(n+1) = bR_j^h(n) + \eta\left(\frac{1}{P}\sum_{\mu=1}^{P} xR1_{\mu j}^h - \frac{1}{P}\sum_{\mu=1}^{P} xR2_{\mu j}^h\right), \ j \in \overline{1, N^h},$$

$$wL_{ij}^{in-h}(n+1) = wL_{ij}^{in-h}(n) + \eta(\rho_{ij}^+ - \rho_{ij}^-), \ i \in \overline{1, N^{in}}, \ j \in \overline{1, N^h},$$

$$\rho_{ij}^+ = \frac{1}{P}\sum_{\mu=1}^{P} x1_{\mu i}^{in} xL1_{\mu j}^h, \ \rho_{ij}^- = \frac{1}{P}\sum_{\mu=1}^{P} x2_{\mu i}^{in} xL2_{\mu j}^h,$$

$$wR_{ij}^{in-h}(n+1) = w_{ij}^{in-h}(n) + \eta(\rho_{ij}^+ - \rho_{ij}^-), \ i \in \overline{1, N^{in}}, \ j \in \overline{1, N^h},$$

$$\rho_{ij}^+ = \frac{1}{P}\sum_{\mu=1}^{P} x1_{\mu i}^{in} xR1_{\mu j}^h, \ \rho_{ij}^- = \frac{1}{P}\sum_{\mu=1}^{P} x2_{\mu i}^{in} xR2_{\mu j}^h,$$

$$wL_{ij}^{out-h}(n+1) = wL_{ij}^{out-h}(n) + \eta(\rho_{ij}^+ - \rho_{ij}^-), \ i \in \overline{1, N^{out}}, \ j \in \overline{1, N^h},$$

$$\rho_{ij}^+ = \frac{1}{P}\sum_{\mu=1}^{P} x1_{\mu i}^{out} xL1_{\mu j}^h, \ \rho_{ij}^- = \frac{1}{P}\sum_{\mu=1}^{P} x2_{\mu i}^{out} xL2_{\mu j}^h$$

$$wR_{ij}^{out-h}(n+1) = w_{ij}^{out-h}(n) + \eta(\rho_{ij}^+ - \rho_{ij}^-), \ i \in \overline{1, N^{out}}, \ j \in \overline{1, N^h},$$

$$\rho_{ij}^+ = \frac{1}{P}\sum_{\mu=1}^{P} x1_{\mu i}^{out} xR1_{\mu j}^h, \ \rho_{ij}^- = \frac{1}{P}\sum_{\mu=1}^{P} x2_{\mu i}^{out} xR2_{\mu j}^h$$

$$wL_{ij}^{h-h}(n+1) = wL_{ij}^{h-h}(n) + \eta(\rho_{ij}^+ - \rho_{ij}^-), \ i \in \overline{1, N^h}, \ j \in \overline{1, N^h},$$

$$\rho_{ij}^+ = \frac{1}{P}\sum_{\mu=1}^{P} xL1_{\mu-1,i}^h xL1_{\mu j}^h, \ \rho_{ij}^- = \frac{1}{P}\sum_{\mu=1}^{P} xL2_{\mu-1,i}^h xL2_{\mu j}^h$$

$$wR_{ij}^{h-h}(n+1) = wR_{ij}^{h-h}(n) + \eta(\rho_{ij}^+ - \rho_{ij}^-), \ i \in \overline{1, N^h}, \ j \in \overline{1, N^h},$$

$$\rho_{ij}^+ = \frac{1}{P}\sum_{\mu=1}^{P} xR1_{\mu+1,i}^h xR1_{\mu,j}^h, \ \rho_{ij}^- = \frac{1}{P}\sum_{\mu=1}^{P} xR2_{\mu+1,i}^h xR2_{\mu j}^h$$

26. If $\frac{1}{P \cdot N^{out}} \sum_{\mu=1}^{P} \sum_{i=1}^{N^{out}} | \, x1_{\mu i}^{out} - x2_{\mu i}^{out} \, | > \varepsilon$, then $n = n+1$, go to 2.

6. Numerical study

The numerical study of the proposed methods for determining the parameter values was carried out in the Google Colaboratory environment using the Tensorflow package.

To determine the structure of a classification model based on RBMRHL with 200 input neurons (corresponding to the number of analyzed words in each text), i.e. determining the number of hidden neurons, a number of experiments were carried out, the results of which are presented in Figure 3.

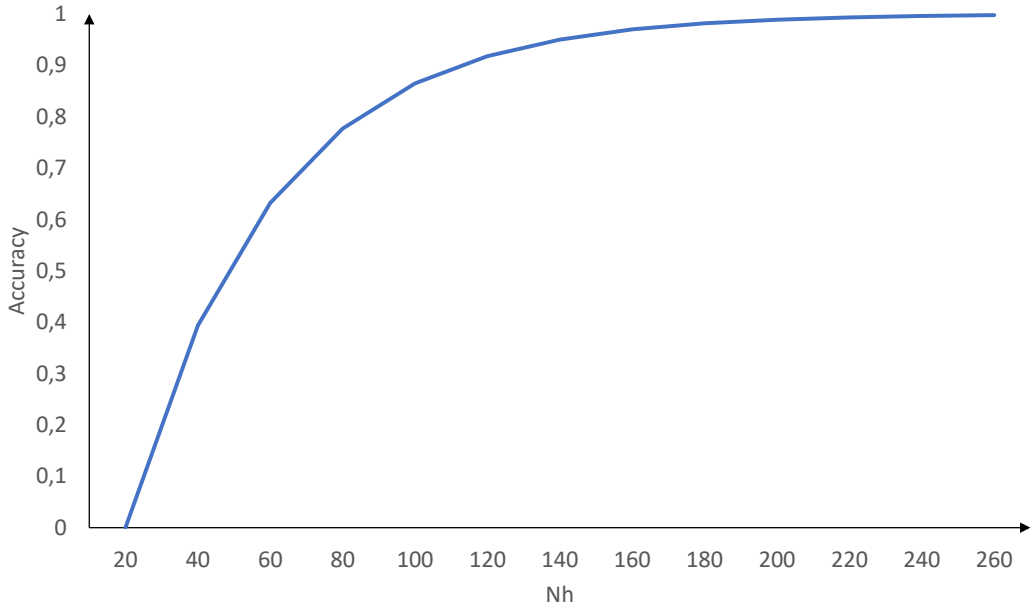

Figure 3: Graph of the dependence of the recognition accuracy value on the number of hidden neurons

The standard IMDB data set was used as the input data to determine the values of the parameters of the neural network classification model. The criterion for choosing the structure of the neural network model was the classification accuracy. As can be seen from the Figure 3, with an increase in the number of hidden neurons, the accuracy value increases. For prediction, it is sufficient to use 200 hidden neurons, since with a further increase in the number of hidden neurons, the change in the accuracy value is insignificant.

Table 1 presents a comparative description of neural networks for recognition, where BRBMRHL means bidirectional RBMRHL.

Table 1

Comparative characteristics of neural networks for recognition

Criterion \ Network	SRN	GRU	LSTM	BRRN	BGRU	BLSTM	RBMRHL/ BRBMRHL
Accuracy	0.80	0.85	0.90	0.92	0.94	0.96	0.91 / 0.98

According to Table 1, BRBMRHL has the highest recognition accuracy.

7. Conclusions

1. To solve the problem of increasing the accuracy of recognition of natural language structures, the existing methods of neural network classification were investigated. These studies have shown that today the most effective is the use of recurrent neural networks.

2. To improve the quality of recognition of natural language constructions, mathematical models of unidirectional and bidirectional stochastic neural networks RBMRHL (Restricted Boltzmann machine with recurrent hidden layer) were created, in which, unlike the traditional RBM (Restricted Boltzmann machine), hidden layer neurons are interconnected.
3. For models of unidirectional and bidirectional stochastic neural networks RBMRHL, methods for identifying their parameters based on contrastive divergence were proposed.
4. In the course of a numerical study of models of unidirectional and bidirectional stochastic neural networks RBMRHL, their structure was determined. The experiments showed that with 200 hidden neurons (corresponding to the number of input neurons), the accuracy value does not change significantly, and the selected network gives recognition results with maximum accuracy.
5. The proposed approach can be used in various intelligent systems that use the recognition of natural language constructs. For example, in supply chain management systems, where natural language interaction between subjects, which are represented by software agents, plays an important role.

8. References

[1] P. F. Dominey, M. Hoen, T. Inui, A neurolinguistic model of grammatical construction processing, in: Journal of Cognitive Neuroscience, vol. 18, issue 12, 2006, pp. 2088–2107. doi: 10.1162/jocn.2006.18.12.2088.

[2] N. Khairova, N. Sharonova, Modeling a logical network of relations of semantic items in super phrasal unities, in: Proceedings of the 2011 9th East-West Design & Test Symposium (EWDTS), 2011, pp. 360-365. doi: 10.1109/EWDTS.2011.6116585.

[3] J. F. Cox, J.G. Schleher, Theory of Constraints Handbook, New York, NY, McGraw-Hill, 2010.

[4] E. M. Goldratt, My saga to improve production, Selected Readings in Constraints Management, Falls Church, VA: APICS (1996) 43-48.

[5] E. M. Goldratt, Production: The TOC Way (Revised Edition) including CD-ROM Simulator and Workbook, Revised edition, Great Barrington, MA: North River Press, 2003.

[6] S. N. Sivanandam, S. Sumathi, S. N. Deepa, Introduction to Neural Networks using Matlab 6.0, The McGraw-Hill Comp., Inc., New Delhi, 2006.

[7] S. Haykin, Neural networks and Learning Machines, Upper Saddle River, New Jersey: Pearson Education, Inc., 2009.

[8] K.-L. Du, K. M. S. Swamy, Neural Networks and Statistical Learning, Springer-Verlag, London, 2014.

[9] E. Fedorov, T. Utkina, O. Nechyporenko, Forecast method for natural language constructions based on a modified gated recursive block, in: CEUR Workshop Proceedings, vol. 2604, 2020, pp. 199-214.

[10] Z. Zhang, Z. Tang, C. Vairappan, A novel learning method for Elman neural network using local search, in: Neural Information Processing – Letters and Reviews, vol. 11, 2007, pp. 181–188.

[11] R. Dey, F. M. Salem, Gate-Variants of Gated Recurrent Unit (GRU) Neural Networks, arXiv:1701.05923, 2017. – URL: https://arxiv.org/ftp/arxiv/papers/1701/1701.05923.pdf.

[12] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 2014, pp. 1724–1734. doi: 10.3115/v1/D14-1179.

[13] H. Jaeger, W. Maass, J. Prıncipe, Special issue on echo state networks and liquid state machines, Neural Networks 20 (2007) 287–289. doi: 10.1016/j.neunet.2007.04.001.

[14] A. H. S. Hamdany, R. R. O. Al-Nima, L. H. Albak, Translating cuneiform symbols using artificial neural network, in: TELKOMNIKA Telecommunication, Computing, Electronics and Control, volume 19, No. 2, 2021, pp. 438-443. doi: 10.12928/telkomnika.v19i2.16134

[15] A. Wysocki and M. Ławryńczuk, Predictive control of a multivariable neutralisation process using Elman neural networks, in: Advances in Intelligent Systems and Computing, Springer:Heidelberg, 2015, pp. 335-344. doi: 10.1007/978-3-319-15796-234.

[16] T. Mikolov, M. Karafi´at, L. Burget, J. Cernock`y, S. Khudanpur, Recurrent neural network based language model, in: 11th Annual Conference of the International Speech Communication Association, 2010, pp. 1045-1048.
[17] M. Sundermeyer, T. Alkhouli, J. Wuebker, H. Ney, Translation modeling with bidirectional recurrent neural networks, in: Proceedings of the Conference on Empirical Methods on Natural Language Processing, 2014, pp. 14-25.
[18] M. Berglund, T. Raiko, M. Honkala, L. Kärkkäinen, A. Vetek, J. Karhunen, Bidirectional recurrent neural networks as generative models, in: Proceedings of the 28th International Conference on Neural Information Processing Systems, 2015, pp. 856−864.
[19] M. Sundermeyer, R. Schluter, H. Ney, LSTM neural networks for language modeling, in: Thirteenth Annual Conference of the International Speech Communication Association, 2012, pp. 194-197. doi: 10.1.1.248.4448.
[20] P. Potash, A. Romanov, A. Rumshisky, Ghostwriter: using an LSTM for automatic rap lyric generation, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 1919– 1924. doi:10.18653/v1/D15-1221.
[21] E. Kiperwasser, Y. Goldberg, Simple and Accurate Dependency Parsing Using Bidirectional LSTM Feature Representations, Transactions of the Association for Computational Linguistics 4 (2016) 313–327. doi: 10.1162/tacl_a_00101.
[22] A. Graves, J. Schmidhuber, Framewise phoneme classification with bidirectional LSTM and other neural network architectures, Neural Networks 18 (2005) 602–610, doi:10.1016/j.neunet.2005.06.042.
[23] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, arXiv preprint arXiv:1412.3555, 2014.
[24] R. Dey, F. M. Salem, Gate-Variants of Gated Recurrent Unit (GRU) Neural Networks, arXiv:1701.05923, 2017. URL: https://arxiv.org/ftp/arxiv/papers/1701/1701.05923.pdf.
[25] S. A. Khan, S. M. D. Khalid, M. A. Shahzad, F. Shafait, Table structure extraction with bi-directional gated recurrent unit networks, in: Proceedings of the 15th International Conference on Document Analysis and Recognition, vol. 4, no. 2, 2019, pp. 78–88. doi:10.1109/ICDAR.2019.00220
[26] G. E. Hinton, A Practical Guide to Training Restricted Boltzmann Machines, Technical Report UTML TR 2010–003, University of Toronto, 2010.
[27] A. Fischer, C. Igel, Training Restricted Boltzmann Machines: An Introduction, Pattern Recognition 47 (2014) 25-39. doi: 10.1016/j.patcog.2013.05.025.