Method of Dynamic Stock Buffer Management Based on a Connectionist Expert System

Eugene Fedorov and Olga Nechyporenko

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

Abstract

The paper proposes a method for dynamic stock buffer management based on a connectionist expert system. The novelty of the study lies in the fact that for the dynamic stock buffer management, a method was created based on the connectionist and logical approach and a neural network model with sigmoid functions for the dynamic stock buffer management. Three criteria for evaluating the effectiveness of the proposed model were selected, the parameters of the proposed model were identified based on the backpropagation method in batch mode, which is focused on the technology of parallel information processing, and the matrix pseudo-inversion method. The proposed model and methods for its parametric identification make it possible to increase the speed, accuracy and reliability of decision making. The proposed method can be used in various intelligent systems.

Keywords

dynamic stock buffer management, theory of constraints, neural network with sigmoid functions, backpropagation method, matrix pseudo-inversion method, CUDA technology

1. Introduction

Nowadays, the creation of computer control systems for various processes is of great importance [1-2]. More and more companies are striving to improve and optimize their business processes based on the implementation of the Theory of Constraints (TOC) technology [3-7], which provides dynamic stock buffer management and is used for supply chain management. As a result, the urgency of developing methods for intellectualizing the theory of constraints technology increases significantly.

The aim of the work is to increase the efficiency of dynamic stock buffer management using a neural network with sigmoid functions, which is trained on the basis of backpropagation and matrix pseudo-inversion methods. To achieve this goal, it is necessary to solve the following tasks:

1. Formation of knowledge about the stock buffer management.
2. Creation of a mathematical model of an artificial neural network with sigmoid functions for dynamic stock buffer management.
3. Selection of criteria for evaluating the effectiveness of a mathematical model of an artificial neural network with sigmoid functions for dynamic stock buffer management.
4. Identification of the parameters of the mathematical model of the artificial neural network with sigmoid functions for dynamic stock buffer management based on the method of backpropagation in batch mode.
5. Identification of the parameters of the mathematical model of the artificial neural network with sigmoid functions for dynamic control of the stock buffer based on knowledge and the method of matrix pseudo-inversion.

2. Literature review

Currently, dynamic stock buffer management is based on production rules [3-7]. There are no computer systems for dynamic stock buffer management that are based on artificial intelligence

Information Technology and Implementation (IT&I-2022), November 30 - December 02, 2022, Kyiv, Ukraine
EMAIL: fedorovee75@ukr.net (E. Fedorov); olne@ukr.net (O. Nechyporenko)
ORCID: 0000-0003-3841-7373 (E. Fedorov); 0000-0002-3954-3796 (O. Nechyporenko)

methods. At the present time, artificial intelligence methods are used to control dynamic objects, with the most popular being artificial neural networks [8-10].

The advantages of neural networks are [11-13]:

- the possibility of their training and adaptation;
- the ability to identify patterns in the data, their generalization, i.e. extracting knowledge from data, so knowledge about the object is not required (for example, its mathematical model);
- parallel processing of information that increases computing power.

The disadvantages of neural networks are [14-16]:

- difficulty in determining the structure of the network, since there are no methods for calculating the number of layers and neurons in each layer for specific applications;
- difficulty in forming a representative sample;
- high probability of the training and adaptation method to hit a local extremum;
- inaccessibility for 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.

Recently, neural networks have been combined with expert systems. The advantage of expert systems is [17-19] the representation of knowledge in the form of association rules that are easily accessible for human understanding. The disadvantages of expert systems are [20-21]:

- the impossibility of their training and adaptation (weights of association rules cannot be automatically adjusted);
- the impossibility of parallel processing of information, which increases computing power.

Thus, it is relevant to create a method for dynamic stock buffer management, which will eliminate these shortcomings.

3. Formation of knowledge about the stock buffer management

It was assumed in the work that the stock buffer is divided into three zones (red, yellow and green) of the same size. The proposed artificial neural network is based on knowledge about the stock buffer management, presented in the form of association rules given below.

1. If the depth of being in the red zone of the stock buffer is at least half of this zone (i.e. the amount of stocks is not more than half of this zone) and staying in the red zone of the stock buffer for at least four days, and staying in the green zone of the stock buffer for at least four days, then check the data, because an anomaly has occurred in them. The conclusion of the rule is encoded as (0,0,0,1)

$$\left(x_1 \le x_1^{\min} + \frac{x_1^{\max} - x_1^{\min}}{6}\right) \wedge (x_2 \ge 4) \wedge (x_3 \ge 4) \xrightarrow{v_1} \mathbf{y} = (0,0,0,1)\,.$$

2. If the depth of being in the red zone of the stock buffer is at least half of this zone (i.e. the amount of stocks is not more than half of this zone) and the stay in the red zone of the stock buffer is at least four days, and the stay in the green zone of the stock buffer is less than four days, then increase the stock buffer size by 1/3. The conclusion of the rule is encoded as (1,0,0,0)

$$\left(x_1 \le x_1^{\min} + \frac{x_1^{\max} - x_1^{\min}}{6}\right) \wedge (x_2 \ge 4) \wedge (x_3 < 4) \xrightarrow{v_2} \mathbf{y} = (1,0,0,0)\,.$$

3. If the depth of being in the red zone of the stock buffer is at least half of this zone (i.e. the amount of stocks is not more than half of this zone) and the stay in the red zone of the stock buffer is less than four days, and the stay in the green zone of the stock buffer is at least four days, then check the data, because an anomaly has occurred in them. The conclusion of the rule is encoded as (0,0,0,1)

$$\left(x_1 \le x_1^{\min} + \frac{x_1^{\max} - x_1^{\min}}{6}\right) \wedge (x_2 < 4) \wedge (x_3 \ge 4) \xrightarrow{v_3} \mathbf{y} = (0,0,0,1)\,.$$

4. If the depth of being in the red zone of the stock buffer is at least half of this zone (i.e. the amount of stocks is not more than half of this zone) and the stay in the red zone of the stock buffer is

less than four days and the stay in the green zone of the stock buffer is less than four days, then increase the stock buffer size by 1/3. The conclusion of the rule is encoded as (1,0,0,0)

$$\left(x_1 \le x_1^{\min} + \frac{x_1^{\max} - x_1^{\min}}{6}\right) \wedge \left(x_2 < 4\right) \wedge \left(x_3 < 4\right) \xrightarrow{v_4} \mathbf{y} = (1,0,0,0)\,.$$

5. If the depth of being in the red zone of the stock buffer is less than half of this zone (i.e. the amount of stocks is more than half of this zone) and staying in the red zone of the stock buffer for at least four days, and staying in the green zone of the stock buffer for at least four days , then check the data, since an anomaly has occurred in them. The conclusion of the rule is encoded as (0,0,0,1)

$$\left(x_1 > x_1^{\min} + \frac{x_1^{\max} - x_1^{\min}}{6}\right) \wedge \left(x_2 \ge 4\right) \wedge \left(x_3 \ge 4\right) \xrightarrow{v_5} \mathbf{y} = (0,0,0,1)\,.$$

6. If the depth of being in the red zone of the stock buffer is less than half of this zone (i.e. the amount of stocks is not more than half of this zone) and the stay in the red zone of the stock buffer is at least four days, and the stay in the green zone of the stock buffer is less than four days , then increase the stock buffer size by 1/3. The conclusion of the rule is encoded as (1,0,0,0)

$$\left(x_1 \le x_1^{\min} + \frac{x_1^{\max} - x_1^{\min}}{6}\right) \wedge \left(x_2 \ge 4\right) \wedge \left(x_3 < 4\right) \xrightarrow{v_6} \mathbf{y} = (1,0,0,0)\,.$$

7. If the depth of being in the red zone of the stock buffer is less than half of this zone (i.e. the amount of stocks is more than half of this zone) and the stay in the red zone of the stock buffer is less than four days, and the stay in the green zone of the stock buffer is at least four days, then decrease the stock buffer size by 1/3. The conclusion of the rule is encoded as (0,1,0,0)

$$\left(x_1 > x_1^{\min} + \frac{x_1^{\max} - x_1^{\min}}{6}\right) \wedge \left(x_2 < 4\right) \wedge \left(x_3 \ge 4\right) \xrightarrow{v_7} \mathbf{y} = (0,1,0,0)\,.$$

8. If the depth of being in the red zone of the stock buffer is less than half of this zone (i.e. the amount of stocks is more than half of this zone) and the stay in the red zone of the stock buffer is less than four days, and the stay in the green zone of the stock buffer is less than four days, then do not change the stock buffer size. The conclusion of the rule is encoded as (0,0,1,0)

$$\left(x_1 > x_1^{\min} + \frac{x_1^{\max} - x_1^{\min}}{6}\right) \wedge \left(x_2 < 4\right) \wedge \left(x_3 < 4\right) \xrightarrow{v_8} \mathbf{y} = (0,0,1,0)\,,$$

where x_1 – current stock size in pieces,

x_2 – time spent in the red zone of the stock buffer in days,

x_3 – time spent in the green zone of the stock buffer in days,

y – action code,

$x_1^{\min}$ – the minimum number of stocks of goods in pieces (the border between the black and red zones of the stock buffer),

$x_1^{\max}$ – the maximum number of stocks of goods in pieces (the border between the green and blue zones of the stock buffer),

v_j – weight of j^{th} association rule.

4. Creation of a mathematical model of the neural network with sigmoid functions for dynamic stock buffer management

To dynamically manage the stock buffer, the work has further improved the mathematical models of the artificial neural network through the use of association rules and multidimensional logistic functions, which reduces the number of hidden layers, which simplifies the identification of artificial neural network parameters. The neural network maps inputs to outputs according to the knowledge of stock buffer management.

The structure of the neural network model with sigmoid functions in the form of a graph is shown in Figure 1.

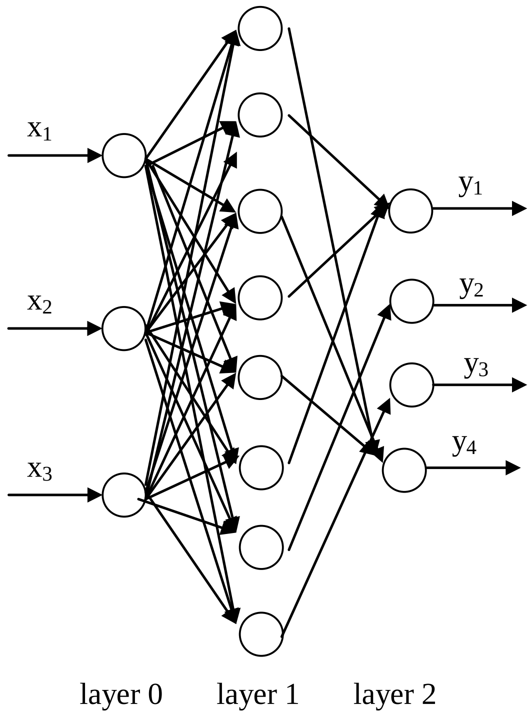

Figure 1: The structure of the neural network model with sigmoid functions in the form of a graph

The input (zero) layer contains three neurons (the number of neurons corresponds to the number of input variables). The hidden layer contains eight neurons (the number of neurons corresponds to the number of association rules). The output layer contains four neurons (the number of neurons corresponds to the number of actions). The functioning of the neural network with sigmoid functions is presented as follows.

In the hidden layer, multidimensional logistic functions are calculated (corresponds to the aggregation of subconditions of association rules connected by a conjunction)

$$y_j^h = f_j(\mathbf{x}) = \prod_{z=1}^{Z} sigm_{zj}(x_z)\,, \quad sigm_{zj}(x_z) = \left(1 + \exp\left[-\left(\frac{x_z - m_{zj}}{\sigma_{zj}}\right)\right]\right)^{-1}, \quad j \in \overline{1, J}\,.$$

Based on the knowledge about stock buffer management, the parameters of the activation functions were chosen as follows:

$$m_{11} = m_{12} = m_{13} = m_{14} = m_{15} = m_{16} = m_{17} = m_{18} = x_1^{\min} + \frac{x_1^{\max} - x_1^{\min}}{6},$$

$$\sigma_{11} = \sigma_{12} = \sigma_{13} = \sigma_{14} = -0.05\,,\ \sigma_{15} = \sigma_{16} = \sigma_{17} = \sigma_{18} = 0.05\,,$$

$$m_{21} = m_{22} = m_{23} = m_{24} = m_{25} = m_{26} = m_{27} = m_{28} = 4\,,\ \sigma_{21} = \sigma_{22} = \sigma_{25} = \sigma_{26} = 0.05\,,$$

$$\sigma_{23} = \sigma_{24} = \sigma_{27} = \sigma_{28} = -0.05\,,\ m_{21} = m_{22} = m_{23} = m_{24} = m_{25} = m_{26} = m_{27} = m_{28} = 4\,,$$

$$\sigma_{31} = \sigma_{33} = \sigma_{35} = \sigma_{37} = 0.05\,,\ \sigma_{32} = \sigma_{34} = \sigma_{36} = \sigma_{38} = -0.05\,.$$

In the output layer, the sums of weighted multivariate logistic functions are calculated (corresponds to the aggregation of activated association rules with the same conclusions, i.e. actions)

$$y_k^{out} = \sum_{j=1}^{J} w_{jk} y_j^h\,,\ k \in \overline{1, K}\,,$$

$$w_{jk}=\begin{cases} v_j, & (j,k)\in\{(2,1),(4,1),(6,1),(7,2),(8,3),(1,4),(3,4),(5,4)\} \\ 0, & (j,k)\notin\{(2,1),(4,1),(6,1),(7,2),(8,3),(1,4),(3,4),(5,4)\} \end{cases}.$$

Thus, the mathematical model of the neural network with sigmoid functions is presented in the form

$$y_k^{out}=\sum_{j=1}^{J} w_{jk}\prod_{z=1}^{Z}\left(1+\exp\left[-\left(\frac{x_z-m_{zj}}{\sigma_{zj}}\right)\right]\right)^{-1},\ k\in\overline{1,K}. \quad (1)$$

To decide on the choice of action for the model (1), the following rule is used

$$k^*=\arg\max_k y_k^{out},\ k\in\overline{1,K}.$$

5. Choice of criteria for evaluating the effectiveness of a mathematical model of a neural network with sigmoidal functions for dynamic stock buffer management

In this work, to evaluate the parametric identification of the mathematical model of the neural network with sigmoidal functions (1), the following were chosen:

- accuracy criterion, which means the choice of such parameters' values $\mathbf{v}=(v_1,...,v_J)$, that deliver a minimum of the mean squared error (the difference between the output according to the model and the desired output)

$$F=\frac{1}{2I}\sum_{i=1}^{I}\sum_{k=1}^{K}(y_{ik}^{out}-d_{ik})^2\to\min_{\mathbf{v}}, \quad (2)$$

where $\mathbf{d}_i=(d_{i1},...,d_{iK})$ – i th test output vector, $d_{ik}\in\{0,1\}$,
$\mathbf{y}_i^{out}=(y_{i1}^{out},...,y_{iK}^{out})$ – output vector obtained from the model,
I – number of test implementations;

- reliability criterion, which means the choice of such parameters' values $\mathbf{v}=(v_1,...,v_J)$, that provide a minimum probability of making an incorrect decision (the difference between the model output and the desired output)

$$F=\frac{1}{I}\sum_{i=1}^{I}\left[\arg\max_{k\in\overline{1,K}} y_{ik}^{out}\neq\arg\max_{k\in\overline{1,K}} d_{ik}\right]\to\min_{\mathbf{v}}, \quad (3)$$

$$\left[\arg\max_{k\in\overline{1,K}} y_{ik}^{out}\neq\arg\max_{k\in\overline{1,K}} d_{ik}\right]=\begin{cases} 1, & \arg\max_{k\in\overline{1,K}} y_{ik}^{out}\neq\arg\max_{k\in\overline{1,K}} d_{ik} \\ 0, & \arg\max_{k\in\overline{1,K}} y_{ik}^{out}=\arg\max_{k\in\overline{1,K}} d_{ik} \end{cases};$$

- performance criterion, which means the choice of such parameters' values $\mathbf{v}=(v_1,...,v_J)$, that deliver a minimum of computational complexity

$$F=T\to\min_{\mathbf{v}}. \quad (4)$$

6. Identification of the parameters of the mathematical model of the neural network with sigmoid functions for dynamic stock buffer management based on the backpropagation method in batch mode

To identify the parameters of the mathematical model of the neural network with sigmoid functions for dynamic stock buffer management (1), the procedure for determining these parameters based on the backpropagation method has been further improved in the work by calculating only the vector of parameters $\mathbf{v}=(v_1,...,v_J)$ and batch learning to speed up learning, which involves the following steps:

1. Initialization by uniform distribution on the interval (0,1) of weights v_j, $z\in\overline{1,Z}$, $j\in\overline{1,J}$.
2. Specifying the training set $\{(\mathbf{x}_i,\mathbf{d}_i)\,|\,\mathbf{x}_i\in R^Z,\mathbf{d}_i\in\{0,1\}^Z\}$, $i\in\overline{1,I}$, where $\mathbf{x}_i$ – i^{th} normalized training input vector, $\mathbf{d}_i$ – i^{th} learning output vector, Z – number of input variables, I – training set cardinality. Specifying parameters for activation functions m_{zj},σ_{zj}, $z\in\overline{1,Z}$, $j\in\overline{1,J}$. Iteration number n=1.
3. Output signal calculation (forward move)

$$y_{ik}^{out}=\sum_{j=1}^{J}w_{jk}\prod_{z=1}^{Z}\left(1+\exp\left[-\left(\frac{x_{iz}-m_{zj}}{\sigma_{zj}}\right)\right]\right)^{-1},\ i\in\overline{1,I},\ k\in\overline{1,K},$$

$$w_{jk}=\begin{cases}v_j, & (j,k)\in\{(2,1),(4,1),(6,1),(7,2),(8,3),(1,4),(3,4),(5,4)\}\\ 0, & (j,k)\notin\{(2,1),(4,1),(6,1),(7,2),(8,3),(1,4),(3,4),(5,4)\}\end{cases}.$$

4. Calculation of error energy based on criterion (3)

$$E=\frac{1}{2I}\sum_{i=1}^{I}\sum_{k=1}^{K}(y_{ik}^{out}-d_{ik})^2.$$

5. Setting the weights of the output layer (backward move)

$$w_{jk}=\begin{cases}w_{jk}-\eta\dfrac{\partial E}{\partial w_{jk}}, & (j,k)\in\{(2,1),(4,1),(6,1),(7,2),(8,3),(1,4),(3,4),(5,4)\}\\ 0, & (j,k)\notin\{(2,1),(4,1),(6,1),(7,2),(8,3),(1,4),(3,4),(5,4)\}\end{cases},\ j\in\overline{1,J},\ k\in\overline{1,K},$$

where η – parameter that determines the learning rate (with a large η learning rate is faster, but the risk of getting the wrong solution increases), $0<\eta<1$,

$$\frac{\partial E}{\partial w_{jk}}=\frac{1}{I}\sum_{i=1}^{I}f_j(\mathbf{x}_i)(y_{ik}^{out}-d_{ik}).$$

6. Check completion condition.

If $n<N$, then increase iteration number n, go to 3.

7. Algorithm for identifying the parameters of the mathematical model of the neural network with sigmoid functions for dynamic stock buffer management based on the method of backpropagation in batch mode

The algorithm for identifying the parameters of the mathematical model of the neural network with sigmoid functions for dynamic stock buffer management based on the backpropagation method in batch mode, designed for implementation on the GPU using CUDA technology, is shown in Figure 2. This block diagram functions as follows.

1. Iteration number n=1, initialization by uniform distribution on the interval (0,1) of weights v_j, $z\in\overline{1,Z}$, $j\in\overline{1,J}$.

2. Specifying the training set $\{(\mathbf{x}_i,\mathbf{d}_i)\,|\,\mathbf{x}_i\in R^Z,\mathbf{d}_i\in\{0,1\}^Z\}$, $i\in\overline{1,I}$, where $\mathbf{x}_p$ – i^{th} normalized training input vector,, $\mathbf{d}_i$ – i^{th} learning output vector, Z – number of input variables, I – the training set cardinality. Specifying parameters for activation functions m_{zj},σ_{zj}, $z\in\overline{1,Z}$, $j\in\overline{1,J}$.

3. Calculation of the output signal according to the model (1), using KI threads, which are grouped into K blocks. Each thread computes y_{ik}^{out}.

4. Calculation of error energy based on criterion (2), using KI strands that are grouped into K blocks. In each block, based on parallel reduction, a partial sum of I elements of the form $\frac{(y_{ik}^{out} - d_{ik})^2}{2I}$ is calculated. The partial sums obtained in each block are added.

5. Adjust the weights of the output layer using JI strands, which are grouped into J blocks. In each block, based on parallel reduction, the sum of I elements of the form $\frac{f_j(\mathbf{x}_i)(y_{ik}^{out} - d_{ik})}{I}$ is calculated.

6. Check completion condition.

If $n < N$, then n=n+1, go to 3.

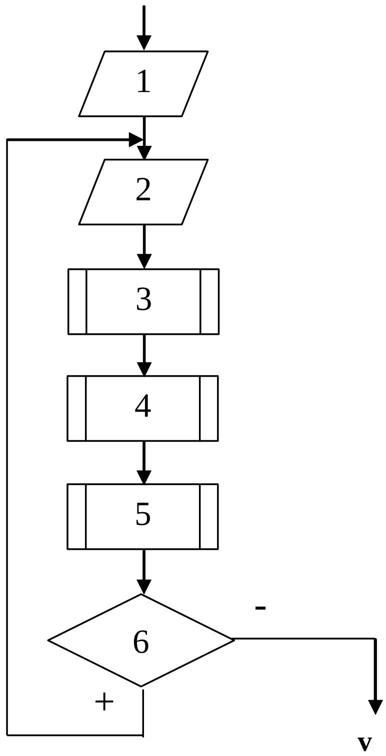

Figure 2: Block diagram of the algorithm for identifying the parameters of the mathematical model of the neural network with sigmoid functions for dynamic stock buffer management based on the backpropagation method in batch mode

8. Identification of the parameters of the mathematical model of the neural network with sigmoid functions for dynamic stock buffer management based on the method of matrix pseudo-inversion

Modification of the output layer weights based on the matrix pseudo-inversion method involves the following steps:

1. Create matrix $\mathbf{Y}^h = [y_{ij}^h]$, $i \in \overline{1, I}$, $j \in \overline{1, J}$.
2. Create matrix $\mathbf{D} = \left[d_{ik}\right]$, $i \in \overline{1, I}$, $k \in \overline{1, K}$.
3. Calculate matrix $\mathbf{W} = [w_{jk}]$, $j \in \overline{1, J}$, $k \in \overline{1, K}$,

$$\mathbf{W} = \left(\mathbf{Y}^h\right)^+ \mathbf{D},$$

where $\left(\mathbf{Y}^h\right)^+$ – pseudo-inverse matrix.

The matrix pseudo-inversion method is based on the singular value decomposition (SVD) method, which allows computing the pseudo-inverse matrix $\left(\mathbf{Y}^h\right)^+$ in the form

$$\left(\mathbf{Y}^h\right)^+ = \mathbf{V}\mathbf{\Sigma}^+ \mathbf{U}^T ,$$

where $\mathbf{\Sigma}^+$ obtained from matrix $\mathbf{\Sigma} = diag(\sigma_1...\sigma_q)$ by replacing each σ_i with $1/\sigma_i$ followed by the transposition of this modified matrix, $q = \min\{J, I\}$.

The Singular Value Decomposition (SVD) method consists of the following steps:

1. The input data matrix is set in the form $\mathbf{Y}^h = [\mathbf{y}_1^h,...,\mathbf{y}_I^h]$, where $\mathbf{y}_i^h$ – vector of dimensions J, $I \geq J$.
2. Bidiagonalization is performed, which uses the Householder transformation (reflection) and allows us to represent matrix $\mathbf{Y}^h$ as a product of matrices

$$\mathbf{Y}^h = \mathbf{U}_1\mathbf{B}\mathbf{V}_1^T$$

or as a procedure call

$$(\mathbf{U}_1, \mathbf{B}_1, \mathbf{V}_1) = \text{bidiagonal}(\mathbf{Y}^h),$$

where matrices $\mathbf{U}_1, \mathbf{V}_1$ are orthogonal and have dimension $J \times J$ and $I \times I$ respectively, matrix $\mathbf{B}_1$ is upper bidiagonal and has dimension $J \times I$.

3. z=1.
4. The QR method is performed, which uses the Givens rotation and allows us to represent matrix $\mathbf{B}$ as a product of matrices

$$\mathbf{B}_z = \mathbf{U}_{z+1}\mathbf{B}_{z+1}\mathbf{V}_{z+1}^T$$

or as a procedure call

$$(\mathbf{U}_{z+1}, \mathbf{B}_{z+1}, \mathbf{V}_{z+1}, e) = \text{qr}(\mathbf{B}_z),$$

where matrices $\mathbf{U}_{z+1}, \mathbf{V}_{z+1}$ are orthogonal and have dimension $J \times J$ and $I \times I$ respectively, matrix $\mathbf{B}_{z+1}$ is upper bidiagonal and has dimension $J \times I$, e – error.

5. If $e > \varepsilon$, then $z = z + 1$, go to step 4.
6. Calculate matrix $\mathbf{U}$

$$\mathbf{U} = \prod_{m=1}^{z} \mathbf{U}_m .$$

7. Calculate matrix $\mathbf{V}$

$$\mathbf{V} = \prod_{m=1}^{z} \mathbf{V}_m .$$

8. Calculate matrix $\mathbf{\Sigma}$

$$\mathbf{\Sigma} = \mathbf{B}_z .$$

Thus, the vector of left singular vectors is represented as $\mathbf{U} = [\mathbf{u}_1...\mathbf{u}_J]$, the vector of right singular vectors is represented as $\mathbf{V} = [\mathbf{v}_1...\mathbf{v}_I]$, the matrix of singular values of dimension $J \times I$ is represented as $\mathbf{\Sigma} = diag(\sigma_1...\sigma_q)$, $q = \min\{J, I\}$.

Bidiagonalization as procedure bidiagonal($\mathbf{X}$*)*

1. Initialization

$$\mathbf{B} = \mathbf{Y}^h,\ \mathbf{B} = [b_{ji}],\ j \in \overline{1,J},\ i \in \overline{1,I},$$

$$\mathbf{U} = \mathbf{I},\ \mathbf{U} = [u_{sj}],\ s \in \overline{1,J},\ j \in \overline{1,J},$$

$$\mathbf{V} = \mathbf{I},\ \mathbf{V} = [v_{iz}],\ i \in \overline{1,I},\ z \in \overline{1,I}.$$

2. z=1.
3. Calculation of the Householder matrix in the form

$$\mathbf{Q}_z = \text{householder}(\mathbf{B}, z, J).$$

4. Calculation of the upper bidiagonal matrix B and the left orthogonal matrix $\mathbf{U}$

$$\mathbf{B} = \mathbf{Q}_z\mathbf{B},\ \mathbf{U} = \mathbf{U}\mathbf{Q}_z.$$

5. If $z \ge \min\{J, I\} - 1$, then go to step 8.
6. Calculation of the Householder matrix in the form

$$\mathbf{P}_z = \text{householder}(\mathbf{B}^T, z+1, I).$$

7. Calculation of the upper bidiagonal matrix B and the right orthogonal matrix $\mathbf{V}_1$

$$\mathbf{B} = \mathbf{B}\mathbf{P}_{z+1},\ \mathbf{V} = \mathbf{P}_{z+1}\mathbf{V}.$$

8. If $z \le \min\{J, I\} - 1$, then $z = z + 1$, go to step 3.

The result is matrices $\mathbf{U}$, $\mathbf{B}$, $\mathbf{V}$.

*Householder transformation as a procedure householder(***B***, l, C)*

1. Vector formation $\mathbf{x} = (x_1, ..., x_C)$ in the form

$$x_c = b_{cl},\ c \in \overline{1, C}.$$

2. Vector formation $\mathbf{u} = (u_l, ..., u_C)$ in the form

$$u_c = \begin{cases} 0, & 1 \le c < l \\ x_c + \operatorname{sgn}(x_c)\sqrt{\sum_{s=c}^{C}(x_s)^2}, & c = l \\ x_c, & l+1 \le c \le C \end{cases}.$$

3. Vector calculation $\mathbf{v} = (v_l, ..., v_C)$ in the form

$$\mathbf{v} = \frac{\mathbf{u}}{\|\mathbf{u}\|}.$$

4. Calculation of the Householder matrix in the form

$$\mathbf{H} = \mathbf{I} - 2\mathbf{v}\mathbf{v}^T.$$

The result is a matrix **H.**

*QR method as procedure qr(***B***)*

1. Initialization

$$\mathbf{\Sigma} = \mathbf{B},\ \mathbf{\Sigma} = [\sigma_{ji}],\ j \in \overline{1, J},\ i \in \overline{1, I},$$

$$\mathbf{U} = \mathbf{I},\ \mathbf{U} = [u_{sj}],\ s \in \overline{1, J},\ j \in \overline{1, J},$$

$$\mathbf{V} = \mathbf{I},\ \mathbf{V} = [v_{iz}],\ i \in \overline{1, I},\ z \in \overline{1, I}.$$

2. l=1.
3. Givens rotation

$$(\cos\theta,\ \sin\theta,\ r) = \text{rot}(\sigma_{ll}, \sigma_{l,l+1}).$$

4. Matrix calculation Q

$$\mathbf{Q} = \mathbf{I},\ \mathbf{Q} = [q_{iz}],\ i \in \overline{1, I},\ z \in \overline{1, I},$$

$$q_{ll} = \cos\theta,\ q_{l,l+1} = \sin\theta,\ q_{l+1,l} = -\sin\theta,\ q_{l+1,l+1} = \cos\theta.$$

5. Calculation of the upper bidiagonal matrix B and the right orthogonal matrix $\mathbf{V}_2$

$$\mathbf{\Sigma} = \mathbf{\Sigma}\mathbf{Q}^T,\ \mathbf{V} = \mathbf{V}\mathbf{Q}^T.$$

6. Zeroing all σ_{ji}, that have $|\sigma_{ji}| \le \varepsilon$.

7. Givens rotation

$$(\cos\theta,\ \sin\theta,\ r) = \mathrm{rot}(\sigma_{ll}, \sigma_{l+1,l}).$$

8. Matrix calculation Q

$$\mathbf{Q} = \mathbf{I},\ \mathbf{Q} = [q_{sj}],\ s \in \overline{1, J},\ j \in \overline{1, J},$$

$$q_{ll} = \cos\theta,\ q_{l,l+1} = \sin\theta,\ q_{l+1,l} = -\sin\theta,\ q_{l+1,l+1} = \cos\theta.$$

9. Calculation of the upper bidiagonal matrix $\mathbf{\Sigma}$ and the left orthogonal matrix $\mathbf{U}_2$

$$\mathbf{\Sigma} = \mathbf{Q\Sigma},\ \mathbf{U} = \mathbf{UQ}^T.$$

10. Zeroing all σ_{ji}, that have $|\sigma_{ji}| \le \varepsilon$.

11. If $l \le \min\{J, I\} - 1$, then $l = l + 1$, go to step 3.

12. Calculation of the error as the sum of the modules of the elements lying above the main diagonal

$$e = \sum_{j=2}^{q} \sum_{i=1}^{j-1} |\sigma_{ji}|,\ q = \min\{J, I\}.$$

The result is matrices $\mathbf{U}$, $\mathbf{\Sigma}$, $\mathbf{V}$ and error *e*.

Givens rotation as a procedure rot(f, g)

Calculates the Givens rotation matrix $\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$, that transforms vector $(f, g)^T$ to vector $(r, 0)^T$, i.e. $\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}\begin{pmatrix} f \\ g \end{pmatrix} = \begin{pmatrix} r \\ 0 \end{pmatrix}$.

If f =0, then $\cos\theta = 0$, $\sin\theta = 1$, $r = g$.

If $|f| > |g|$ and $f \ne 0$, then $\tan\theta = \dfrac{g}{f}$,

$$\cos\theta = \frac{1}{\sqrt{1 + (\tan\theta)^2}} = \frac{1}{\sqrt{1 + \left(\frac{g}{f}\right)^2}},\quad \sin\theta = \frac{\tan\theta}{\sqrt{1 + (\tan\theta)^2}} = \frac{g}{f} \cdot \frac{1}{\sqrt{1 + \left(\frac{g}{f}\right)^2}},\quad r = f\cos\theta = \frac{f}{\sqrt{1 + \left(\frac{g}{f}\right)^2}}.$$

If $|f| \le |g|$ and $f \ne 0$, then $\mathrm{cotan}\ \theta = \dfrac{f}{g}$,

$$\sin\theta = \frac{1}{\sqrt{1 + (\mathrm{cotan}\,\theta)^2}} = \frac{1}{\sqrt{1 + \left(\frac{f}{g}\right)^2}},\quad \cos\theta = \frac{\mathrm{cotan}\,\theta}{\sqrt{1 + (\mathrm{cotan}\,\theta)^2}} = \frac{f}{g} \cdot \frac{1}{\sqrt{1 + \left(\frac{f}{g}\right)^2}},\quad r = f\sin\theta = \frac{g}{\sqrt{1 + \left(\frac{f}{g}\right)^2}}.$$

The result is the elements $\cos\theta$, $\sin\theta$, *r*.

9. Experiments and results

A numerical study of the proposed mathematical model of a neural network with sigmoid functions and a conventional multilayer perceptron was carried out in the Matlab package using the Deep Learning Toolbox (to identify the parameters of the model of a multilayer perceptron and a neural network with radial basis functions based on backpropagation, as well as to identify the parameters of the proposed neural network model with sigmoid functions (1) based on back propagation and matrix pseudo-inversion).

Table 1 presents the computational complexity, root mean square errors (RMSE), the probabilities of making wrong decisions on the dynamic stock buffer management, obtained on the basis of the data set of the logistics company "Ekol Ukraine" and "Vitronic" (This dataset contains 2000 precedents. During the simulation, 80 % of the precedents were randomly selected for training, and 20 % for testing procedures) using an artificial neural network such of the multilayer perceptron (MLP) type and radial basic function neural networks (RBFNN) with back propagation (BP), as well as the proposed model (1) with back propagation (BP) and matrix pseudo-inversion. At the same time, MLP had 2 hidden layers (each consisted of 6 neurons, like the input layer), RBFNN had one hidden layer of 12 neurons. I – is the training set cardinality, N – is the number of iterations performed.

According to Table 1, the best results in terms of computational complexity are given by model (1) with parameter identification based on BP in batch mode using CUDA (reduces the computational complexity by I times), and the best results in terms of RMSE and the probability of making an incorrect decision are given by model (1) with parameter identification based on matrix pseudo-inversion without using CUDA.

Table 1
Computational complexity, root mean square error, probability of wrong decisions on dynamic stock buffer management

Model and method of parameter identification	RMSE	Probability of making the wrong decision	Computational complexity
MLP with BP with logistic activation function without using CUDA	0.50	0.20	*T=IN*
RBFNN with BP in batch mode with Gaussian activation function without using CUDA	0.40	0.15	*T=IN*
Author's model (1) with BP in batch mode using CUDA	0.10	0.07	*T=N*
Author's model (1) with BP with matrix pseudo-inversion without using CUDA	0.05	0.02	*T=IN*

10. Conclusions

1. To solve the problem of improving the efficiency of dynamic stock buffer management, relevant artificial intelligence methods have been investigated. These studies have shown that by far the most effective is the use of artificial neural networks in combination with expert systems.
2. The novelty of the study lies in the fact that the proposed method of dynamic stock buffer management is based on logic and artificial neural networks. It provides representation of knowledge about stock buffer management in the form of association rules, reduces computational complexity, mean squared error, and the probability of making the wrong decision by automatically choosing the model structure, reducing the probability of hitting a local extremum, and using CUDA parallel information processing technology.
3. As a result of the numerical study, it was found that the proposed method of dynamic stock buffer management based on the connectionist expert system in the case of identifying the parameters of the neural network model based on the method of matrix pseudo-inversion, provides the probability of making the wrong decision on the dynamic stock buffer management of 0.02, and the root-mean-square error of 0.05, and in the case of identifying the parameters of the neural network model based on the backpropagation method in batch mode using CUDA, it reduces the computational complexity by a factor of I, where I is the training set cardinality.
4. Further research prospects are the use of the proposed method of dynamic stock buffer management based on the connectionist expert system for various intelligent systems for managing dynamic objects in natural language.

11. References

[1] G. G. Shvachych, O. V. Ivaschenko, V. V. Busygin, Ye. Ye. Fedorov, Parallel computational algorithms in thermal processes in metallurgy and mining, Naukovyi Visnyk Natsionalnoho Hirnychoho Universytetu, 4 (2018) 129–137. doi: 10.29202/nvngu/2018-4/19.

[2] G. Shlomchak, G. Shvachych, B. Moroz, E. Fedorov, D. Kozenkov, Automated control of temperature regimes of alloyed steel products based on multiprocessors computing systems, Metalurgija, 58 (2019) 299-302.

[3] U. P. Nagarkatte, N. Oley, Theory of constraints and thinking processes for creative thinkers: creative problem solving, Boca Raton, Fl, CRC Press, 2018.

[4] B. Sproull, Theory of Constraints, Lean, and Six Sigma Improvement Methodology: Making the Case for Integration, London, CRC Press, 2019.

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

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

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

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

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

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

[11] E. Fedorov, V. Lukashenko, V. Patrushev, A. Lukashenko, K. Rudakov, S. Mitsenko, The method of intelligent image processing based on a three-channel purely convolutional neural network, in: CEUR Workshop Proceedings, vol. 2255, 2018, pp. 336–351. doi: 10.1109/EWDTS.2013.6673185

[12] . 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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

[17] P. B. Shekhawat, S. S. Dhande, Building an iris plant data classifier using neural network associative classification, international journal of advancements in technology, 2 (2011) 491-506.

[18] P. B. Shekhawat, S. S. Dhande, A classification technique using associative classification, International journal of computer application, 20 (2011) 20-28.

[19] M. A. Kadhim, A. Alam, H. Kaur, Design and implementation of intelligent agent and diagnosis domain tool for rule-based expert system, in: International Conference on Machine Intelligence and Research Advancement, 2013, pp. 619-622. doi: 10.5815/ijisa.2016.09.08.

[20] S. Ticketek , O. Abdoun, J. Abouchabaka, An expert system for a constrained mobility management of human resources, in: 10th International Colloquium on Logistics and Supply Chain Management (LOGISTIQUA), Rabat, Morocco, 2017, pp. 53-58.

[21] A. Asemi, A. Ko, M. Nowkarizi, Intelligent libraries: a review on expert systems, artificial intelligence, and robot, Library Hi Tech, 39 (2021), 412-434. doi:10.1108/lht-02-2020-0038.