ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

UDC 612.741.1:519.876.5 **doi: 10.32620/reks.2023.2.05**

Eugene FEDOROV, Olga NECHYPORENKO, Maryna CHYCHUZHKO, Vladyslav CHYCHUZHKO, Ruslan LESHCHENKO

Cherkasy State Technical University, Cherkasy, Ukraine

NEURAL NETWORK-BASED METHODS FOR FINDING THE SHORTEST PATH AND ESTABLISHING ASSOCIATIVE CONNECTIONS BETWEEN OBJECTS

Nowadays, solving optimizations problems is one of the tasks for intelligent computer systems. Currently, there is a problem of insufficient efficiency of optimizations tasks solving methods (for example, high computing time and/or accuracy). The object of the research is the process of finding the shortest path and establishing associative connections between objects. The subject of the research is the methods of finding the shortest path and establishing associative connections between objects based on neural networks with associative memory and neural network reinforcement training. The objective of this work is to improve the efficiency of finding the shortest path and establishing associative connections between objects through neural networks with associative memory and neural network reinforcement training. To achieve this goal, a neuro-associative method and a neural network reinforcement training method was developed. The advantages of the proposed methods include the following. First, the proposed bi-directional recurrent correlative associative memory, which uses hetero-associative and auto-associative memory and an exponential weighting function, allows for increasing the associative memory capacity while preserving learning accuracy. Second, the Deep Q-Network (DQN) reinforcement learning method with dynamic parameters uses the ε-greedy approach, which in the initial iterations is close to random search, and in the final iterations is close to directed search, which is ensured by using dynamic parameters and allows increasing the learning speed while preserving learning accuracy. Conducted numerical research allowed us to estimate both methods (for the first method, the root mean square error was 0.02, and for the second method it was 0.05). The proposed methods allow expanding the field of application of neural networks with associative memory and neural network reinforcement learning, which is confirmed by their adaptation for the tasks of finding the shortest path and establishing associative connections between objects and contribute to the effectiveness of intelligent computer systems of general and special purpose. Prospects for further research are to investigate the proposed methods for a wide class of artificial intelligence problems.

***Keywords**: reinforcement learning; neural network; associative memory; establishing associative connections between objects; finding the shortest path.*

Introduction

Motivation

Nowadays, the relevant task is to develop methods aimed at solving problems of finding the shortest path and establishing associative connections between objects, which are used in intelligent computer systems of general and special purpose. An application of such methods may be to determine the location of a product in a warehouse based on its characteristics and to find the shortest path to it.

Accurate methods for solving problems of finding the shortest path and establishing associative connections between objects have high computational complexity. Methods based on local search have a high probability of hitting the local extremum. Methods based on random search do not guarantee convergence. This raises the problem of insufficient efficiency of methods problems of finding the shortest path and establishing associative connections between objects, which needs to be solved.

Current status

At present, reinforcement learning is actively used [3, 4] to solve optimization problems by finding the shortest path in addition to metaheuristic methods [1, 2]. The main directions of single-agent learning with reinforcement are:

– dynamic programming [5, 6];
– adaptive dynamic programming [7, 8];
– Monte-Carlo [9, 10];
– Temporal-Difference Learning [11, 12];
– policy-based methods [13, 14];
– actor-critic methods [15, 16].

There is also multiagent reinforcement learning, which usually uses temporal-difference learning.

Existing reinforcement learning methods have one or more disadvantages:

– a priori knowledge needed about the probabilities of transitions between states and a priori knowledge about the rewards for transitions between states because of action [17, 18];

– there is no way to directly optimize the action selection policy and the cost function calculating policy [19, 20];

– a large number of interactions between the agent and the environment [21, 22];

– converges to a global optimum only in the case of a finite number of actions and states [23, 24];

– susceptible to undertraining [20, 21];

– susceptible to overtraining [22, 24];

– update the value of the cost function only after receiving the entire trajectory (a sequence of state-action-reward triplets) [19, 20];

– requires several long trajectories [17, 21].

This raises the challenge of building effective reinforcement learning methods.

At present, methods based on associative neural networks play an important role in solving problems and establishing associative connections between objects [25, 26]. Methods based on associative neural networks can be divided into methods with associative or hetero-associative memory [27].

Existing neuro-associative methods have one or more of the following disadvantages:

1. Do not have an associative memory [28, 29].
2. Do not have hetero-associative memory [30, 31].
3. Do not work with floating point data [32, 33].
4. Do not have a high capacity for associative memory [30, 34].
5. Not very accurate [29, 30].
6. Have high computational complexity.

This raises the challenge of building effective neuro-associative methods.

Objectives

The object of research. The process of finding the shortest path and establishing associative connections between objects.

The subject of research. Methods for finding the shortest path and establishing associative connections between objects based on neural networks with associative memory and neural network learning with reinforcement.

The purpose of this work is to improve the efficiency of finding the shortest path and establishing associative connections between objects through neural networks with associative memory and neural network learning with reinforcement.

To achieve this goal, it is necessary to solve the following tasks:

1. Create a neural network method and establishing associative connections between objects.

2. Develop a neural network learning method with reinforcement to find the shortest path.

3. Conduct a numerical study of the proposed optimization methods.

1. Problem statement

The problem of improving the efficiency of establishing associative connections between objects and solving the problem of the shortest path is reduced to the problem of finding a vector of parameters W , that satisfies the criterion of the adequacy of the artificial neural network based on associative memory and neural network model of the state-action cost function and represented in the form

$$F = \frac{1}{P}\sum_{\mu=1}^{P}(f(x_\mu, W) - d_\mu)^2 \to \min_W ,$$

i.e., deliver a minimum of the root-mean-square error (the difference between the model output and the desired output), where P – test set power, x_μ – μ-th training input value, d_μ .– μ-th training output value

2. Neural network-based method for establishing associative connections between objects

Variable generation

Characteristics of objects (parameters) - these are distinctive properties and characteristics of objects, quantitative and qualitative data about them.

For example, in the case of a warehouse the following main parameters were selected as input variables characterizing the goods: article, brand, name, product grouping, size, volume, weight, composition, shelf life, price, consumer characteristics, and quantity.

In the case of a warehouse as input variables that characterize the location of goods, may be chosen the following basic parameters of goods in the warehouse: sector, cell, container, shelf, box. Requirements for specifying characteristics:

– the minimum number of characteristics for a product – 3;

– the main characteristics necessary for the selection and identification of goods must be specified;

– it is obligatory to indicate belonging to a class, group, subgroup, type of goods;

– characteristics are filled out in the same language;

– the maximum allowed number of characters for the characteristic value – 255 (including spaces).

Input and output variables are represented in bipolar or binary form.

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

Block diagram of a neural network model of bidirectional recurrent correlated associative memory

Figure 1 shows a block diagram of the bidirectional recurrent correlation associative memory proposed in the work (BRCAM), which is a recurrent artificial neural network (ANN) with one hidden layer.

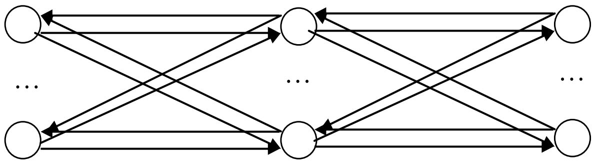

Fig. 1. Bidirectional recurrent correlative associative memory (BRCAM)

Implements hetero-associative memory (its element is represented by a pair of patterns $(\mathbf{m}_x, \mathbf{m}_y)$, $\mathbf{m}_y \neq \mathbf{m}_x$) and auto-associative memory (its element is represented by a pair of samples $(\mathbf{m}_x, \mathbf{m}_y)$, $\mathbf{m}_y = \mathbf{m}_x$) and restores the memorized sample pair $(\mathbf{m}_x, \mathbf{m}_y)$ on a key sample $\mathbf{m}_x$, corresponding to the first vector $\mathbf{x}$, or by a key sample $\mathbf{m}_y$, corresponding to the second vector $\mathbf{d}$.

The basic BRCAM options are as follows:

1. Correlation matrix associative memory (CMAM) with linear weighting function $f(s) = s$, which is similar to the Hopfield neural network. Memory capacity (number of pairs $(\mathbf{m}_x, \mathbf{m}_y)$) is $\frac{N}{2\ln N}$ or 0.14N.

2. High-order correlative associative memory (HOCAM) with a step function of weighting $f(s) = s^a$. The memory capacity (number of pairs $(\mathbf{m}_x, \mathbf{m}_y)$) is N^a

3. Exponential correlative associative memory (ECAM) with indicative weighing function $f(s) = a^s$. Memory capacity (number of pairs $(\mathbf{m}_x, \mathbf{m}_y)$) in the case of $a = 2$ or $N > 1 + a^2$ is 2^N.

The most important property of BRCAM is that the same ANN with the same link weights can store and reproduce several memorized samples.

BRCAM has the following advantages:

1. Used to restore sample pairs.

2. Provides a good generalization quality (the submitted sample may be noisy or have missing information).

3. No need to determine the number of hidden layers (one hidden layer).

4. Does not require determining the number of hidden layer neurons (coincides with the number of training samples).

5. Parameter identification is done in one iteration, so the parameter and structure adaptation is fast.

This study proposes a discrete bi-directional recurrent correlative associative memory (DBRCAM) and a continuous bi-directional recurrent correlation associative memory (CBRCAM).

DBRCAM and CBRCAM training uses one-step learning.

CBRCAM uses Leaky Integrate and Fire (LIF) neurons,

The LIF neuron model is represented as

$$C \cdot R \frac{du}{dt} = -u(t) + I(t)R$$

or

$$\tau \frac{du}{dt} = -u(t) + I(t)R \ ,$$

where τ – time constant,

C – capacity,

R – resistance,

u(t) –voltage,

I(t) –amperage.

The identification of ANN DBRCAM and CBRCAM model parameters (storage phase)

A training set is specified $\{(\mathbf{x}_j, \mathbf{d}_j) \mid \mathbf{x}_j \in \{-1,1\}^{N^x}, \mathbf{d}_j \in \{-1,1\}^{N^y}\}$, $j \in \overline{1, N^{(1)}}$, P – training set power. Weights of forward and backward connections are initialized $w_{ij} = x_{ji}$, $i \in \overline{1, N^x}$, $j \in \overline{1, N^{(1)}}$, $v_{ji} = d_{ji}$, $i \in \overline{1, N^y}$, $j \in \overline{1, N^{(1)}}$, where N^x – sample length $\mathbf{m}_x$, where N^y – sample length $\mathbf{m}_y$, $N^{(1)}$ – the number of neurons in the hidden layer.

Note. In the case of binary data $x_{\mu j} = 2x_{\mu j} - 1$, $d_{\mu j} = 2d_{\mu j} - 1$.

Model of ANN DBRCAM operation (recovery phase) (by vector $\mathbf{x}$ restoring the $\mathbf{m}_x$ sample and $\mathbf{m}_y$ sample)

1. $z_i(1) = x_i$ (if $\mathbf{x}$ bipolar) or $z_i(1) = 2x_i - 1$ (if $\mathbf{x}$ binary), $i \in \overline{1, N^x}$, $n = 1$.

2. $s_j(n+1) = f(\sum_{i=1}^{N^x} w_{ij} z_i(n))$, $j \in \overline{1, N^{(1)}}$,

where $f(s)$ – weighing function.

3. $y_i(n+1) = \mathrm{sgn}(\sum_{j=1}^{N^{(1)}} v_{ji} s_j(n+1))$, $i \in \overline{1, N^y}$.

4. $s_j(n+1) = f(\sum_{i=1}^{N^y} v_{ji} y_i(n+1))$, $j \in \overline{1, N^{(1)}}$.

5. $z_i(n+1) = \mathrm{sgn}(\sum_{j=1}^{N^{(1)}} w_{ij} s_j(n+1))$, $i \in \overline{1, N^x}$.

6. If $\sum_{i=1}^{N^x} | z_i(n+1) - z_i(n) | > 0$, then $n = n+1$, move to step 2.

The results are $\mathbf{m}_y = (y_1, ..., y_{N^y})$ and $\mathbf{m}_x = (z_1, ..., z_{N^x})$ samples.

Model of ANN DBRCAM operation (recovery phase) (by vector $\mathbf{d}$ restoring the $\mathbf{m}_x$ sample and $\mathbf{m}_y$ sample)

1. $y_j(1) = d_j$ (if $\mathbf{d}$ bipolar) or $y_j(1) = 2d_j - 1$ (if $\mathbf{d}$ binary), $j \in \overline{1, N^y}$, $n = 1$.

2. $s_j(n+1) = f(\sum_{i=1}^{N^y} v_{ji} y_i(n))$, $j \in \overline{1, N^{(1)}}$,

where $f(s)$ – weighing function.

3. $z_i(n+1) = \mathrm{sgn}(\sum_{j=1}^{N^{(1)}} w_{ij} s_j(n+1))$, $i \in \overline{1, N^x}$.

4. $s_j(n+1) = f(\sum_{i=1}^{N^x} w_{ij} z_i(n))$, $j \in \overline{1, N^{(1)}}$.

5. $y_i(n+1) = \mathrm{sgn}(\sum_{j=1}^{N^{(1)}} v_{ji} s_j(n+1))$, $i \in \overline{1, N^y}$.

6. If $\sum_{j=1}^{N^y} | y_j(n+1) - y_j(n) | > 0$, then $n = n+1$, move to step 2.

The results are $\mathbf{m}_y = (y_1, ..., y_{N^y})$ and $\mathbf{m}_x = (z_1, ..., z_{N^x})$ samples.

Model of ANN CBRCAM functioning (recovery phase) (by vector $\mathbf{x}$ restoring the $\mathbf{m}_x$ sample and $\mathbf{m}_y$ sample)

1. $z_i(\Delta t) = x_i$ (if $\mathbf{x} \in [-1,1]^{N^x}$) or $z_i(\Delta t) = 2x_i - 1$ (if $\mathbf{x} \in [0,1]^{N^x}$), $i \in \overline{1, N^x}$, $s_j(\Delta t) = 0$, $j \in \overline{1, N^{(1)}}$, $t = \Delta t$.

2.

$$s_j(t+\Delta t) = \left(1 - \frac{\Delta t}{\tau}\right) s_j(t) + \frac{\Delta t}{\tau} \left(\sum_{i=1}^{N^x} w_{ij} \phi(z_i(t)) \right) R,$$

$$j \in \overline{1, N^{(1)}},$$

$\phi(s)$=tanh(s),

where Δt – quantization step, $0 < \Delta t < 1$,

τ – time constant (usually is 1),

R – resistance (usually is 1),

ϕ – activation function.

3.

$$y_i(t+\Delta t) = \left(1 - \frac{\Delta t}{\tau}\right) y_i(t) + \frac{\Delta t}{\tau} \left(\sum_{j=1}^{N^{(1)}} v_{ji} f(s_j(t+1)) \right) R,$$

$$i \in \overline{1, N^y},$$

where $f(s)$ – weighing function.

4.

$$s_j(t+\Delta t) = \left(1 - \frac{\Delta t}{\tau}\right) s_j(t) + \frac{\Delta t}{\tau} \left(\sum_{i=1}^{N^y} v_{ji} \phi(y_i(t)) \right) R,$$

$$j \in \overline{1, N^{(1)}}.$$

5.

$$z_i(t+\Delta t) = \left(1 - \frac{\Delta t}{\tau}\right) z_i(t) + \frac{\Delta t}{\tau} \left(\sum_{j=1}^{N^{(1)}} w_{ij} f(s_j(t)) \right) R,$$

$$i \in \overline{1, N^x}.$$

6. If $\sum_{i=1}^{N^x} | z_i(t+\Delta t) - z_i(t) | > \varepsilon$, then $t = t + \Delta t$, move to step 2.

7. $z_i = \mathrm{sgn}(\phi(z_i(t+\Delta t)))$, $i \in \overline{1, N^x}$.

The results are $\mathbf{m}_y = (y_1, ..., y_{N^y})$ and $\mathbf{m}_x = (z_1, ..., z_{N^x})$ samples.

Model of ANN CBRCAM functioning (recovery phase) (by vector d restoring the $\mathbf{m}_x$ sample and $\mathbf{m}_y$ sample)

1. $z_i(\Delta t) = d_i$ (if $\mathbf{d} \in [-1,1]^{N^y}$) or $z_i(\Delta t) = 2x_i - 1$ (if $\mathbf{d} \in [0,1]^{N^y}$), $i \in \overline{1, N^y}$, $s_j(\Delta t) = 0$, $j \in \overline{1, N^{(1)}}$, $t = \Delta t$.

2. $$s_j(t+\Delta t) = \left(1 - \frac{\Delta t}{\tau}\right) s_j(t) + \frac{\Delta t}{\tau}\left(\sum_{i=1}^{N^y} v_{ji}\phi(y_i(t))\right) R,$$

$j \in \overline{1, N^{(1)}}$,

$\phi(s) = \tanh(s)$,

where Δt – quantization step, $0 < \Delta t < 1$,

τ – time constant (usually is 1),

R – resistance (usually is 1),

ϕ – activation function

3.
$$z_i(t+\Delta t) = \left(1 - \frac{\Delta t}{\tau}\right) z_i(t) + \frac{\Delta t}{\tau}\left(\sum_{j=1}^{N^{(1)}} w_{ij} f(s_j(t+1))\right) R$$

, $i \in \overline{1, N^x}$,

where $f(s)$ – weighing function.

4.
$$s_j(t+\Delta t) = \left(1 - \frac{\Delta t}{\tau}\right) s_j(t) + \frac{\Delta t}{\tau}\left(\sum_{i=1}^{N^x} w_{ij}\phi(z_i(t))\right) R,$$

$j \in \overline{1, N^{(1)}}$.

5.
$$y_i(t+\Delta t) = \left(1 - \frac{\Delta t}{\tau}\right) y_i(t) + \frac{\Delta t}{\tau}\left(\sum_{j=1}^{N^{(1)}} v_{ji} f(s_j(t))\right) R,$$

$i \in \overline{1, N^y}$.

6. If $\sum_{i=1}^{N^y} | y_i(t+\Delta t) - y_i(t) | > \varepsilon$, then $t = t + \Delta t$, move to step 2.

7. $y_i = \operatorname{sgn}(\phi(y_i(t+\Delta t)))$, $i \in \overline{1, N^y}$.

The results are $\mathbf{m}_y = (y_1, ..., y_{N^y})$ and $\mathbf{m}_x = (z_1, ..., z_{N^x})$ samples.

3. A neural network learning method with reinforcement to find the shortest path

Consider the problem of finding the shortest path as the problem of finding the shortest path in a rectangular world of tiles (Tileworld). All cells of this world are squares of the same size and can be tiles or obstacles (traps/pits are not used). Moving is possible only on tiles, and from the current tile you can move only to the tile of its neighborhood, which is the Neumann neighborhood (Fig. 2). A tile can be visited no more than once. In the world of tiles there is one source tile and one target tile. Find the shortest path from the source tile to the target tile.

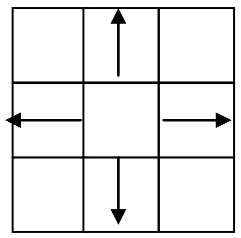

Fig. 2. Example of all admissible motions on Neumann neighborhood tiles

Let a multilayer perceptron of the following form be chosen as a model of the state-action cost function

$$z_j^{(0)} = s_j, \; j \in \overline{1, S^{(0)}},$$

$$z_k^{(l)} = f^{(l)}\left(b_k^{(l)} + \sum_{j=1}^{S^{(l-1)}} w_{jk}^{(l)} z_j^{(k-1)}\right), \; k \in \overline{1, S^{(l)}}, l \in \overline{1, L-1},$$

$$y_k = z_k^{(L)} = b_k^{(L)} + \sum_{j=1}^{S^{(L-1)}} w_{jk}^{(L)} z_j^{(L-1)}, \; k \in \overline{1, S^{(L)}},$$

where $S^{(l)}$ – the number of neurons in the l layer,

L – number of layers,

$b_k^{(l)}$ – offsets (thresholds),

$w_{kj}^{(l)}$ – weights,

$f^{(l)}$ – activation function.

Usually, they are limited to two hidden layers, and the number of neurons in the hidden layer coincides with the number of neurons in the input layer.

For the DQN method with dynamic parameters the neural network can be represented as

$$y_k = Q_\theta(\mathbf{s}, k), \; k \in \overline{1, S^{(L)}},$$

where $\theta = (b_1^{(1)}, ..., b_{S^{(L)}}^{(L)}, w_{11}^{(1)}, ..., w_{S^{(L-1)}S^{(L)}}^{(L)})$ – vector of parameters.

For this method, each state corresponds to the cell of the tile world in which the agent is.

Applied to the world of tiles:

$S^{(0)}$ – the number of cells of the tile world, $S^{(0)} = |S|$, $|S| = \text{height} * \text{width}$,

$S^{(L)}$ – the number of possible actions of an agent, $S^{(L)} = |A|$,

$\mathbf{s}$ – a binary state vector with one 1 (1 corresponds to *s*),

$\mathbf{y}$ – vector of values of the state-action cost function,

$height$ – the height of the tile world in cells,

$width$ – the width of the tile world in cells,

$Q_\theta(\mathbf{s}, k)$ – state-action cost function.

The method consists of the following steps:

1. Initializing.

1.1. Initializing neural network parameter vector θ by means of uniform distribution on the intervals (0,1) or [-0.5, 0.5].

1.2. We specify a discrete set of states (cells) S.

1.3. A discrete set of actions is specified A.

1.4. Parameters are set $\varepsilon^{\min}, \varepsilon^{\max}$ for ε-greedy approach, $0 < \varepsilon^{\min} < \varepsilon^{\max} < 1$, parameters $\gamma^{\min}, \gamma^{\max}$ (determine the importance of future rewards), $0 < \gamma^{\min} < \gamma^{\max} < 1$.

1.5. The reward is initialized $R(s,a)$.

For example, the following rules are used:

– if the agent has moved to a tile cage, then $R(s,a) = 0$;

– if the agent moved to the target cell, then $R(s,a) = 1$.

2. Iteration number $n = 1$.

3. The parameters are calculated

$$\varepsilon(n) = \varepsilon^{\max} - (\varepsilon^{\max} - \varepsilon^{\min})\frac{n-1}{N-1},$$

$$\gamma(n) = \gamma^{\min} + (\gamma^{\max} - \gamma^{\min})\frac{n-1}{N-1}.$$

4. Time moment number $t = 1$.

5. The initial state is observed (cell) s_t, which is converted into an initial state vector $\mathbf{s}_t$.

6. Selects an action a_t for the transition from the state vector $\mathbf{s}_t$, using the ε-greedy policy (if $U(0,1) < \varepsilon(n)$, where $U(0,1)$ – a function that returns a uniformly distributed random number in the range [0,1], then select the action a_t randomly from a set of actions A, else $a_t = \arg\max_{c \in A} Q_\theta(\mathbf{s}_t, c)$).

7. Remuneration $R(s_t, a_t)$ is observed and a new state (cell) s_{t+1}, which is converted into a vector of a new state $\mathbf{s}_{t+1}$.

For example, in the case of four agent actions (up, down, left, right) the following rules are used for the Neumann neighborhood:

– if the action a_t corresponds to the upward movement and the state s_t does not match the cell in the first line of the map, then the agent goes to the state of s_{t+1}, which corresponds to the cell at the top s_t;

– if the action a_t corresponds to the downward movement and the state s_t does not match the cell in the last line of the map, then the agent enters the state of s_{t+1}, which corresponds to the cell from below s_t;

– if the action a_t corresponds to the movement to the left and the state s_t does not match the cell in the first column of the map, then the agent enters the state of s_{t+1}, which corresponds to the cell on the left s_t;

– if the action a_t corresponds to the movement to the right and the state s_t does not match the cell in the first column of the map, then the agent enters the state of s_{t+1}, which corresponds to the cell on the right s_t.

8. The Q target value is calculated:

– if an agent has moved into the tile cage, then, using a greedy policy

$$G_\theta(s_t, a_t) = R(s_t, a_t) + \gamma(n)\max_{c \in A} Q_\theta(\mathbf{s}_{t+1}, c);$$

– if the agent moved to the target cell, then

$$G_\theta(s_t, a_t) = R(s_t, a_t).$$

9. The vector of state-action cost function values is calculated

$$d_{tk} = \begin{cases} G_\theta(s_t, a_t), & k = a_t \\ Q_\theta(\mathbf{s}_t, k), & k \neq a_t \end{cases}, \quad k \in A.$$

10. A neural network is training (a vector of parameters is determined θ) on one training epoch (e.g., based on the gradient descent method), with the training input data being the $\mathbf{s}_t$ vector, and the training output being the $\mathbf{d}_t$ vector

$$\theta = \theta + \eta\nabla_\theta\left(\mathbf{d}_t - Q_\theta(s_t, a_t)\right)^2.$$

11. If the cell to which the agent moved is not the target cell, then $t = t + 1$, move to step 6.

12. If the current iteration is not the last iteration, i.e. $n < N$, then increase the iteration number, i.e. $n = n + 1$, move to step 3, stop otherwise.

4. Experiments

Numerical investigation of the proposed methods was performed using the Python package.

For the reinforcement learning method, the value of the parameters $\varepsilon^{\min} = 0.1, \varepsilon^{\max} = 0.9$ (manage the ε-greedy policy), parameters $\gamma^{\min} = 0.1, \gamma^{\max} = 0.9$ (manage discounting).

Dependence of the parameter $\gamma(n)$ is defined as $\gamma(n) = \gamma^{\min} + (\gamma^{\max} - \gamma^{\min})\frac{n-1}{N-1}$ and is shown in Fig. 3.

Dependence of the parameter $\varepsilon(n)$ is defined as $\varepsilon(n) = \varepsilon^{\max} - (\varepsilon^{\max} - \varepsilon^{\min})\frac{n-1}{N-1}$ and is presented in Fig. 4.

The results of the comparison of the proposed bi-directional recurrent correlation memory (BRCAM) with bi-directional associative memory (BAM) based on the mean square error criterion and capacity on the standard TIMIT database are presented in Table 1

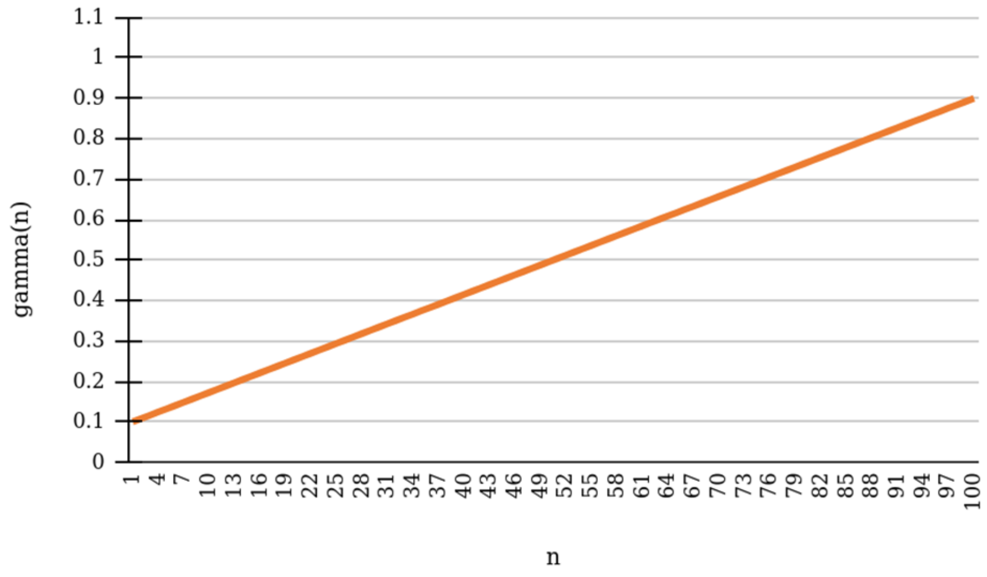

Fig. 3. Dependence of the parameter $\gamma(n)$ on the iteration number n

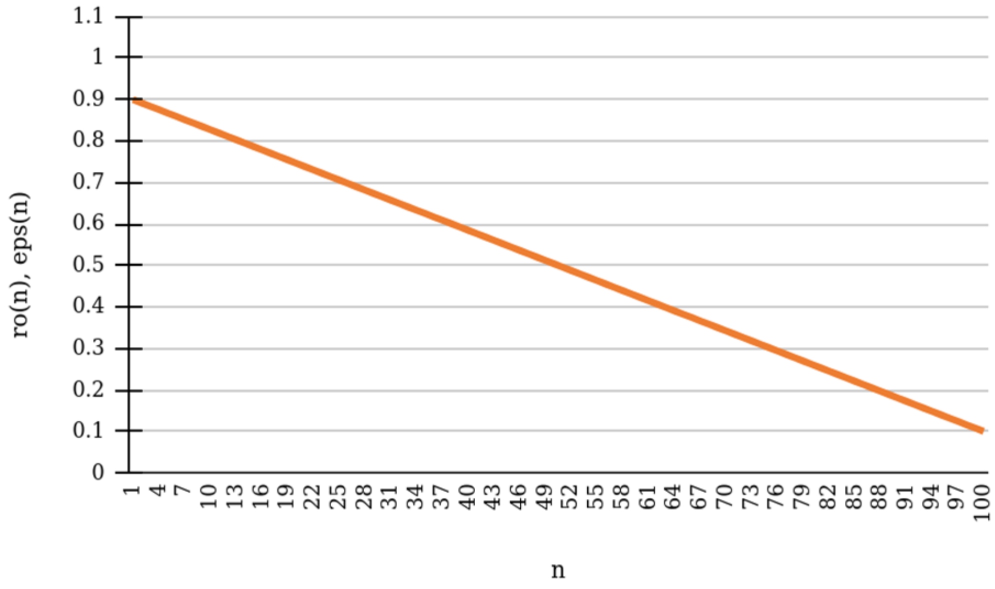

Fig. 4. Dependence of parameters $\varepsilon(n)$ from the iteration number n

Table 1
Comparison of the proposed BRCAM neural network with a traditional BAM neural network

F (Root-mean-square error of the method)		Capacity	
BRCAM	BAM	BRCAM	BAM
0.02	0.02	2^N	$\frac{N}{2\ln N}$

The results of the comparison of the proposed reinforcement learning method DQN with dynamic parameters and the traditional DQN method based on the mean square error criterion and the number of iterations on the standard database https://digitalcommons.du.edu/gridmaps2D, which is described in [31], are presented in Table 2.

Table 2
Comparison of the proposed method with the traditional DQN method

F (Root-mean-square error of the method)		Number of iterations	
suggested	current	suggested	current
0.05	0.05	420	4050

5. Discussion of the results

Advantages of the proposed methods:

1. The proposed bi-directional recurrent correlative associative memory (BRCAM), which uses hetero-associative and auto-associative memory and an exponential weighting function, allows increasing the capacity of associative memory while maintaining learning accuracy (see Table 1).

2. The reinforcement learning method DQN with dynamic parameters uses ε-greedy approach, which in the initial iterations is close to random search, and in the final iterations is close to directed search. This is ensured by the use of dynamic parameters and allows increasing the learning speed while maintaining the learning accuracy (see Table 2).

The dependence (see Fig. 3) of the parameter $\gamma(n)$ on the iteration number n shows that its share increases with increasing iteration number.

The dependence (see Fig. 4) of the parameter $\varepsilon(n)$ on the iteration number n shows that its share decreases with increasing iteration number

6. Applications

The proposed bi-directional recurrent correlation memory (BRCAM) method, applied to the standard database TIMIT, provided inventory determination with a mean square error of 0.02 at a capacity of 2^N (see Table 1). A specific feature of this database is its focus on objects with several characteristics, an example of which would be goods in the warehouse.

The proposed reinforcement learning method DQN with dynamic parameters, applied to the standard database https://digitalcommons.du.edu/gridmaps2D, ensured finding the optimal path to the stock of goods in the warehouse with an average square error of 0.05 with 420 iterations (see Table 2). A specific feature of this database is its focus on the cellular world, an example of which would be a warehouse of goods.

Future studies will investigate the proposed methods not only on standard databases but also on the warehouse.

Conclusions

1. To solve the problem of insufficient efficiency of finding the shortest path, the existing methods of statistical and machine learning were investigated. These studies have shown that by far the most effective are neural network methods.

2. Modification of the recurrent correlative associative memory allowed using this neural network for hetero-associative memory, and the indicative weighting function allows increasing the capacity of associative memory, which stores the characteristics of objects and their location.

3. The modification of the DQN method using dynamic parameters makes it possible to increase the speed of finding the shortest path.

Prospects for further research are to investigate the proposed methods for large inventory warehouses.

Contributions of authors: conceptualization – **Eugene Fedorov**, **Olga Nechyporenko**; methodology – **Eugene Fedorov**, **Olga Nechyporenko**; formulation of tasks – **Eugene Fedorov**, **Olga Nechyporenko**; development of model – **Eugene Fedorov**, **Olga Nechyporenko**; software – **Vladyslav Chychuzhko**, **Ruslan Leshchenko**; verification **- Maryna Chychuzhko**; analysis of results – **Eugene Fedorov**, **Olga Nechyporenko**; visualization **- Eugene Fedorov**; writing – original draft preparation – **Ruslan Leshchenko**; writing – review and editing – **Maryna Chychuzhko**.

References

1. Neskorodieva, T., Fedorov, E., Chychuzhko, M. & Chychuzhko, V. Metaheuristic method for searching quasi-optimal route based on the ant algorithm and annealing simulation. *Radioelectronic and Computer Systems*, 2022, no. 1, pp. 92-102. DOI: 10.32620/reks.2022.1.07.
2. Strilets, V., Donets, V., Ugryumov, M., Artiuch, S., Zelenskyi, R. & Goncharova. T. Agent-oriented data clustering for medical monitoring. *Radioelectronic and Computer Systems*, 2022, no 1, pp. 103-114. DOI: 10.32620/reks.2022.1.08.
3. Ottoni, A. L. C., Nepomuceno, E. G., de Oliveira, M. S. & de Oliveira, D. C. R. Reinforcement learning for the traveling salesman problem with refueling. *Complex & Intelligent Systems*, 2021, vol. 8, pp. 2001-2015. DOI: 10.1007/s40747-021-00444-4.
4. Jaderberg, M. *Human-level performance in first-person multiplayer games with population-based deep reinforcement learning.* Available at: https://arxiv.org/abs/1807.01281. (accessed Feb. 15, 2023).
5. Silver, D. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXivLabs*, 2017. DOI: 10.48550/arXiv.1712.01815.
6. Silver, D. et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 2018, no. 362, iss. 6419, pp. 1140-1144. DOI: 10.1126/science.aar6404.
7. Pouncy, T., Gershman, S., Tsividis, P., Xu, J. & Tenenbaum, J. *Human learning in Atari.* Available at: http://hdl.handle.net/1721.1/112620 (accessed Feb. 15, 2023).
8. Kumaran, D., Hassabis, D. & McClelland, J. What learning systems do intelligent agents need? Complementary learning systems theory updated. *Trends Cogn. Sci*, 2016, no. 20, iss. 7, pp. 512-534. DOI: 10.1016/j.tics.2016.05.004.
9. Gershman, S. J. & Daw, N. D. Reinforcement learning and episodic memory in humans and animals: an integrative framework. *Annu. Rev. Psychol*, 2017, no. 68, pp. 101-128. DOI: 10.1146/annurev-psych-122414-033625.
10. Ballard, I., Wagner, A. D. & McClure, S. M. Hippocampal pattern separation supports reinforcement learning. *bioRxiv*, 2018, pp. 293-332. DOI: 10.1101/293332.
11. Vikbladh, O., Shohamy, D. & Daw, N. Episodic contributions to model-based reinforcement learning. *Annual Conference on Cognitive Computational Neuroscience CCN*, 2017, pp. 1-2. Available at: https://ccneuro.org/abstracts/abstract_3000200.pdf. (accessed Feb. 15, 2023).
12. Kulkarni, T. D., Narasimhan, K., Saeedi, A. & Tenenbaum, J. Hierarchical deep reinforcement learning: integrating temporal abstraction and intrinsic motivation. *arXivLabs*, 2016, pp. 1-14. DOI: 10.48550/arXiv.1604.06057.
13. Higgins, I. et al. Darla: improving zero-shot transfer in reinforcement learning. *arXivLabs*, 2018. DOI: 10.48550/arXiv.1707.08475.
14. Botvinick, M., Weinstein, A., Solway, A. & Barto, A. Reinforcement learning, efficient coding, and the statistics of natural tasks. *Current Opinion in Behavioral Sciences*, 2015, vol. 5, pp. 71-77. DOI: 10.1016/j.cobeha.2015.08.009.
15. Botvinick, M., Ritter, S., Wang, J. X., Kurth-Nelson, Z., Blundell, C. & Hassabis, D. Reinforcement learning, fast and slow. *Trends in Cognitive Sciences*, 2019, vol. 23, iss. 5, pp. 408-422. DOI: 10.1016/j.tics.2019.02.006.
16. Hessel, M. Modayil, J., Hasselt, H. et al. Rainbow: combining improvements in deep reinforcement learning. *arXivLabs*, 2017. DOI: 10.48550/arXiv.1710.02298.
17. Duan, Y., Schulman, J., Chen, X., Bartlett, P.L., Sutskever, I. & Abbeel, P. Rl^2: Fast reinforcement learning via slow reinforcement learning. *arXivLabs*, 2017. DOI: 10.48550/arXiv.1611.02779.
18. Dunovan, K. & Verstynen, T. Believer-skeptic meets actor-critic: Rethinking the role of basal ganglia pathways during decision-making and reinforcement learning. *Frontiers in Neuroscience*, 2016, vol. 10. DOI: 10.3389/fnins.2016.00106.
19. Sutton, R. S. & Barto, A. G. *Reinforcement Learning: An Introduction (2nd ed.).* The MIT Press, 2018. 338 p. Available at: https://inst.eecs.berkeley.edu/~cs188/fa19/assets/files/SuttonBartoIPRLBook2ndEd.pdf (accessed Feb. 15, 2023).
20. Wang, J. X., Kurth-Nelson, Z., Kumaran, D., Tirumala, D., Soyer, H. & Leibo, J. Z. Prefrontal cortex as a meta-reinforcement learning system. *Nature Neuroscience*, 2018, vol. 21, iss. 66, pp. 860-868. DOI: 10.1038/s41593-018-0147-8.
21. Wang, J. X., Kurth-Nelson, Z., Tirumala, D., Soyer, H., Leibo, J. Z. & Munos. R. Learning to reinforcement learn. *arXivLabs*, 2017, pp.*1611.05763.*
22. Xu, Z., Hasselt, H., Hessel, M., Oh, J., Singh, S. & Silver, D. Meta-Gradient Reinforcement Learning with an Objective Discovered Online. *arXivLabs*, 2020. DOI: 10.48550/arXiv.2007.08433.
23. Xu, Z., Hasselt, H. & Silver, D. Meta-gradient reinforcement learning. *arXivLabs*, 2018. DOI: 10.48550/arXiv.1805.09801.
24. Robertazzi, F., Vissani, M., Schillaci, G. & Falotico, E. Brain-inspired meta-reinforcement learning cognitive control in conflictual inhibition decision-making task for artificial agents. *Neural Networks*, 2022,

vol. 154, pp. 283-302. DOI: 10.1016/j.neunet. 2022.06.020.

25. Lobo, R. A. & Valle, M. E. Ensemble of Binary Classifiers Combined Using Recurrent Correlation Associative Memories. *arXivLabs,* 2020, pp. 1-14. DOI: 10.48550/arXiv.2009.08578.

26. Krivtsov, S., Meniailov, I., Bazilevych, K. & Chumachenko. D. Predictive model of COVID-19 epidemic process based on neural network. *Radioelectronic and Computer Systems*, 2022, no. 4, pp. 7-18. DOI: 10.32620/reks.2022.4.01.

27. Martyniuk, T., Krukivskyi, B., Kupershtein, L. & Lukichov V. Neural network model of heteroassociative memory for the classification task. *Radioelectronic and Computer Systems,* 2022, no. 2, pp. 108-117. DOI: 10.32620/reks.2022.2.09.

28. Kobayashi, M. Quaternionic Hopfield neural networks with twin-multistate activation function. *Neurocomputing,* 2017, vol. 267, pp. 304-310. DOI: 10.1016/j.neucom.2017.06.013.

29. Javidmanesh, E. Global Stability and Bifurcation in Delayed Bidirectional Associative Memory Neural Networks with an Arbitrary Number of Neurons. *J. Dyn. Sys., Meas., Control,* 2017, vol. 139, iss. 8. 5 p. DOI: 10.1115/1.4036229.

30. Du, K. L. & Swamy, M. N. S. *Neural Networks and Statistical Learning.* Springer-Verlag London, 2014, 824 p. DOI: 10.1007/978-1-4471-5571-3.

31. Aggarwal, C. C. *Neural Networks and Deep Learning.* Cham, Switzerland, Springer, 2018. 497 p. DOI: 10.1007/978-3-031-29642-0.

32. Neskorodieva, T. & Fedorov, E. Method for automatic analysis of compliance of settlements with suppliers and settlements with customers by neural network model of forecast. *Advances in Intelligent Systems and Computing, AISC,* Springer, Cham, 2021, vol. 1265, pp. 156–165. DOI: 10.1007/978-3-030-58124-4_15.

33. Neskorodieva, T. & Fedorov, E. Neural Network models ensembles for generalized analysis of audit data transformations. *Lecture Notes in Networks and Systems,* Springer, Cham, 2022, vol. 344, pp. 263-279. DOI: 10.1007/978-3-030-89902-8_21.

34. Sturtevant, N. R. Benchmarks for Grid-Based Pathfinding. *IEEE Transactions on Computational Intelligence and AI in Games,* 2012, vol. 4, no. 2, pp. 144-148. DOI: 10.1109/TCIAIG.2012.2197681.

Received 03.04.2023, Accepted 20.05.2023

НЕЙРОМЕРЕЖЕВІ МЕТОДИ ПОШУКУ НАЙКОРОТШИХ ШЛЯХІВ ТА ВСТАНОВЛЕННЯ АСОЦІАТИВНИХ ЗВ’ЯЗКІВ МІЖ ОБ’ЄКТАМИ

Євген Федоров, Ольга Нечипоренко, Марина Чичужко,
Владислав Чичужко, Руслан Лещенко

На сьогоднішній день для інтелектуальних комп'ютерних систем загального та спеціального призначення актуальним є розв'язування оптимізаційних задач. Нині існує проблема недостатньої ефективності методів розв'язання оптимізаційних задач (наприклад, великий час та/або точність обчислень). Об'єктом дослідження є процес пошуку найкоротших шляхів та встановлення асоціативних зв’язків між об'єктами. Предметом дослідження є методи пошуку найкоротших шляхів та встановлення асоціативних зв’язків між об'єктами на основі нейромереж з асоціативною пам'яттю та нейромережевого навчання з підкріпленням. Метою роботи є підвищення ефективності пошуку найкоротших шляхів та встановлення асоціативних зв’язків між об'єктами за рахунок нейромереж з асоціативною пам'яттю та нейромережевого навчання з підкріпленням. Для досягнення поставленої мети в роботі було створено нейро-асоціативний метод та розроблено метод нейромережевого навчання з підкріпленням. До переваг запропонованих методів можна віднести наступні. По-перше, запропонована двоспрямована двонаправлена рекурентна кореляційна кореляційна асоціативна пам'ять, що використовує гетероасоціативну та автоасоціативну пам'ять і показову функцію зважування, дає змогу підвищити місткість асоціативної пам'яті за збереження точності навчання. По-друге, у методі навчання з підкріпленням DQN з динамічними параметрами використовується ε-жадібний підхід, який на початкових ітераціях близький до випадкового пошуку, а на заключних ітераціях близький до направленого пошуку, що забезпечується використанням динамічних параметрів і дає змогу підвищити швидкість навчання за умови збереження точності навчання. Проведене чисельне дослідження дало змогу оцінити обидва методи (для першого методу середньоквадратична помилка становила 0,02, а для другого методу становила 0,05). Запропоновані методи дають змогу розширити сферу застосування нейромереж з асоціативною пам'яттю та нейромережевого навчання з підкріпленням, що підтверджується їхньою адаптацією до задач встановлення асоціативних зв’язків між об'єктами та пошуку найкоротших шляхів, та сприяють підвищенню ефективності інтелектуальних комп'ютерних систем загального та спеціального призначення. Перспективи подальших досліджень полягають у дослідженні запропонованих методів для широкого класу задач штучного інтелекту.

Ключові слова: навчання з підкріпленням; нейромережа; асоціативна пам'ять; встановлення асоціативних зв'язків між об'єктами; пошук найкоротшого шляху.

Федоров Євген Євгенович – д-р техн. наук, проф., проф. каф. робототехніки та спеціалізованих комп'ютерних систем, Черкаський державний технологічний університет, Черкаси, Україна.

Нечипоренко Ольга Володимирівна – канд. техн. наук, доц., доц. каф. робототехніки та спеціалізованих комп'ютерних систем, Черкаський державний технологічний університет, Черкаси, Україна.

Чичужко Марина Володимирівна – канд. техн. наук, доц., доц. каф. робототехніки та спеціалізованих комп'ютерних систем, Черкаський державний технологічний університет, Черкаси, Україна.

Чичужко Владислав Олегович – асп., Черкаський державний технологічний університет, Черкаси, Україна.

Лещенко Руслан Сергійович – асп., Черкаський державний технологічний університет, Черкаси, Україна.

Eugene Fedorov – Doctor of Technical Science, Professor of Department of Robotics and specialized computer systems, Cherkasy State Technological University, Cherkasy, Ukraine,
e-mail: fedorovee75@ukr.net, ORCID: 0000-0003-3841-7373, Scopus Author ID: 47161087200, 47161155900, 57205185819

Olga Nechyporenko – PhD in Engineering Science, Associate Professor of Department of Robotics and specialized computer systems, Cherkasy State Technological University, Cherkasy, Ukraine,
e-mail: olne@ukr.net, ORCID: 0000-0002-3954-3796, Scopus Author ID 57216488854.

Maryna Chychuzhko – PhD in Engineering Science, Associate Professor of Department of Robotics and specialized computer systems, Cherkasy State Technological University, Cherkasy, Ukraine,
e-mail: ckc@chdtu.edu.ua, ORCID: 0000-0001-5329-7897.

Vladyslav Chychuzhko – PhD student of Department of Robotics and specialized computer systems, Cherkasy State Technological University, Cherkasy, Ukraine,
e-mail: ckc@chdtu.edu.ua.

Ruslan Leshchenko – PhD student of Department of Robotics and specialized computer systems, Cherkasy State Technological University, Cherkasy, Ukraine,
e-mail: ckc@chdtu.edu.ua.