Analysis of multi-step algorithms for cognitive maps learning

A. JASTRIEBOW\(^1,2\) and K. POCZĘTA\(^1\)*

\(^1\) Department of Computer Science Applications, Kielce University of Technology,
7 Tysiąclecia Państwa Polskiego Aven., 25-314 Kielce, Poland
\(^2\) Institute of Computer Science, Kazimierz Pulaski University of Technology and Humanities in Radom,
20A Malczewskiego St., 26-600 Radom, Poland

Abstract. This article is devoted to the analysis of multi-step algorithms for cognitive maps learning. Cognitive maps and multi-step supervised learning based on a gradient method and unsupervised one based on the non-linear Hebbian algorithm were described. Comparative analysis of these methods to one-step algorithms, from the point of view of the speed of convergence of a learning algorithm and the influence on the work of the decision systems was performed. Simulation results were done on prepared software tool ISEMK. Obtained results show that implementation of the multi-step technique gives certain possibilities to get quicker values of target relations values and improve the operation of the learned system.

Key words: cognitive maps, multi-step learning algorithm, gradient method, Hebbian algorithm.

1. Introduction

The concept of multi-step algorithms (also known as methods with adequate memory) was introduced in optimization of statistic objects, and then analyzed in many works connected with the parametric adaptation (identification) of dynamic systems [1]. Later, methods of similar kind were introduced in the algorithms area of supervised learning of artificial neuron networks type MLP (e.g. the backpropagation method with the momentum is a typical two-step algorithm of supervised learning [2]) and cognitive maps (CM) [3, 4]. In this paper comparative analysis of multi-step methods to one-step algorithms, from the point of view of the speed of convergence of a learning algorithm and the influence on the work of the decision systems (stability of the systems and an average percentage prediction error obtained during testing) was performed.

The basis of the structure of a cognitive map is a directed graph, which nodes denote concepts significant for the researched phenomenon. The concepts influence each other with an intensity described by the weights of connections (relations) between them, taking on the values from the range [-1,1]. CM can be initialized based on expert knowledge [5, 6] or create automatically based on real data [7]. The crucial issue connected with the analysis of CM is their ability to improve their operation on the light of experience (learning of the relations matrix). Adaptation of the relations matrix weight can be done by unsupervised learning based on the Hebbian method [8–10] and supervised ones with the use of evolutionary computation [7, 11] or a gradient method [3, 4]. A cognitive map is therefore flexible, easy to use and transparent tool for modeling complex dynamic decision-making systems [3–9, 12–14].

This article is devoted to the analysis of a multi-step algorithms for CM learning, which are some kind of generalization of known one-step methods, which wide review was shown in [8]. Learning of cognitive maps was based on a reference map and real data, taken from the UCI Machine Learning Repository. Simulation research together with the analysis results were done on the prepared software tool ISEMK (Intelligent Expert System based on Cognitive Maps) [15]. Part 2 briefly characterizes the structure and a dynamic model of CM. Part 3 presents the multi-step algorithms of CM. Part 4 describes the ISEMK system, enabling the synthesis and the analysis of CM. Part 5 presents selected results of simulation research of the developed algorithms, done in ISEMK. Part 6 contains a summary of the thesis.

2. Cognitive maps

The basis of the structure of CM is a directed graph in the form [16]:

\[
\langle X, R \rangle,
\]

where \(X = [X_1, \ldots, X_n]^T\) – the set of the concepts, \(n\) – the number of concepts; \(X_i(t)\) – the value of the \(i\) concept; \(R = \{r_{j,i}\}\) – relations matrix, \(r_{j,i}\) – the relation weight between the concept \(j\) and the concept \(i\).

In this work a nonlinear dynamics model of a cognitive map described by the relation (2) was used [16, 17]:

\[
X_i(t + 1) = F \left( X_i(t) + \sum_{j=1}^{n} r_{j,i} \cdot X_j(t) \right),
\]

where \(F(x)\) – stabilizing function; \(t\) – discrete time.

Figure 1 presents an exemplary structure of a cognitive map.

\*e-mail: k.piotrowska@tu.kielce.pl
The values of the concepts are calculated until a cognitive map reaches one of the following states [18]:

- a stable state or an unstable state (fixed point attractor, limit cycle),
- a chaotic state (chaotic attractor).

Stability analysis of (2) is very important, especially for modeling complex dynamic systems with lots of relations between concepts.

CM have the ability to learn the relations matrix $R$ based on expert knowledge and real data. Below the idea of multi-step learning algorithms is presented.

3. Multi-step learning algorithms

Characteristic feature of multi-step algorithms is the estimation of a current value of the weight $r_{j,i}$ on the basis of a few previous estimations, which can be achieved according to the Eq. (3) [1, 3]:

$$r_{j,i}(t + 1) = P_{[-1,1]} \left( \sum_{k=0}^{m_1} \alpha_k \cdot r_{j,i}(t - k) + \sum_{l=0}^{m_2} \beta_l \cdot \eta(t) \cdot \Delta J_{j,i}(t - l) \right),$$

where $\alpha_k$, $\beta_l$, $\eta$ - learning parameters ($k = 1, \ldots, m_1$; $l = 1, \ldots, m_2$), $m_1$, $m_2$ - the number of the steps of the method; $\Delta J_{j,i}(t)$ - pseudogradient of selected function of the learning error; $P_{[-1,1]}$ - operator design for the set [-1,1].

Supervised learning based on a gradient method is a modification of the weights in the direction of steepest descent of an error function described by the formula [3]:

$$J(t) = \frac{1}{n} \sum_{i=1}^{n} (X_i(t) - X_i(t - 1))^2,$$

where $Z_i(t)$ – the reference value of the $i$ concept.

The gradient of the function (4) describes equation:

$$\Delta J_{j,i}(t) = (Z_i(t) - X_i(t)) \cdot y_{j,i}(t),$$

where $y_{j,i}(t)$ – sensitivity function, describes as follows:

$$y_{j,i}(t + 1) = (y_{j,i}(t) + X_j(t)) \cdot F'(X_i(t) + \sum_{j=1}^{n} X_j(t) \cdot r_{j,i}(t)).$$

$$J(t) = \frac{1}{n} \sum_{i=1}^{n} (X_i(t) - X_i(t - 1))^2.$$

The pseudogradient for the NHL algorithm describes Eq. (8):

$$\Delta J_{j,i}(t) = X_i(t) \cdot (X_j(t) - \text{sgn}(r_{j,i}(t)) \cdot r_{j,i}(t) \cdot X_i(t)).$$

Simulation analysis of multi-step algorithms of supervised learning, based on a gradient method and unsupervised ones, based on the NHL method was realized based on ISEMK software tool. The basic functionality of ISEMK is described below.

4. ISEMK system

ISEMK is a universal tool for modeling of complex decision-making systems based on CM. ISEMK realizes:

- the implementation of cognitive maps based on expert knowledge and real data,
- supervised and unsupervised learning by multi-step algorithms,
- dynamic monitoring and testing of CM,
- proper visualizations of done research on devised computer software.

Figure 2 represents an exemplary visualization of testing the operation of learned map in the ISEMK system.

5. Selected results of simulation analysis

This part presents selected results of comparative analysis of multi-step methods to one-step algorithms, from the point of view of:

- the speed of convergence of a learning algorithm based on a reference map and the gradient method type (3), (5) (an educational example),
- stability of the map learned with using a gradient method type (3), (5) and real data for the valuation of housing,
- an average percentage prediction error of the map learned based on gradient method type (3), (5), NHL algorithm type (3), (8) and real data for the diagnosis of hepatitis.
5.1. Speed of convergence of learning algorithm. In order to analyze the influence of the steps number \(m_1, m_2\) on the speed of convergence of a learning algorithm, a cognitive map was initialized with random values and then learned based on a reference map using the gradient method type (3), (5). The structure of the initialized map is presented in Fig. 3.

![Fig. 3. Structure of the initialized map](image)

Below the relations matrix for the reference map (Table 1), for the map learned using one-step gradient method (Table 2) and multi-step gradient method (Table 3) are presented.

Table 1

<table>
<thead>
<tr>
<th>(X_1)</th>
<th>(X_2)</th>
<th>(X_3)</th>
<th>(X_4)</th>
<th>(X_5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(X_1)</td>
<td>0</td>
<td>−0.09</td>
<td>0</td>
<td>0.01</td>
</tr>
<tr>
<td>(X_2)</td>
<td>0.12</td>
<td>0</td>
<td>0.08</td>
<td>−0.12</td>
</tr>
<tr>
<td>(X_3)</td>
<td>0</td>
<td>0.16</td>
<td>0</td>
<td>0.09</td>
</tr>
<tr>
<td>(X_4)</td>
<td>0.06</td>
<td>−0.1</td>
<td>−0.08</td>
<td>0</td>
</tr>
<tr>
<td>(X_5)</td>
<td>0</td>
<td>0.18</td>
<td>−0.14</td>
<td>−0.04</td>
</tr>
</tbody>
</table>

Table 2

<table>
<thead>
<tr>
<th>(X_1)</th>
<th>(X_2)</th>
<th>(X_3)</th>
<th>(X_4)</th>
<th>(X_5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(X_1)</td>
<td>0</td>
<td>−0.08</td>
<td>−0.1</td>
<td>0</td>
</tr>
<tr>
<td>(X_2)</td>
<td>0.12</td>
<td>0</td>
<td>−0.16</td>
<td>0.13</td>
</tr>
<tr>
<td>(X_3)</td>
<td>0.04</td>
<td>0.11</td>
<td>0</td>
<td>−0.04</td>
</tr>
<tr>
<td>(X_4)</td>
<td>0.01</td>
<td>−0.02</td>
<td>0.03</td>
<td>0</td>
</tr>
<tr>
<td>(X_5)</td>
<td>0</td>
<td>−0.08</td>
<td>0.03</td>
<td>−0.17</td>
</tr>
</tbody>
</table>

Table 3

<table>
<thead>
<tr>
<th>(X_1)</th>
<th>(X_2)</th>
<th>(X_3)</th>
<th>(X_4)</th>
<th>(X_5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(X_1)</td>
<td>0</td>
<td>−0.06</td>
<td>−0.08</td>
<td>0</td>
</tr>
<tr>
<td>(X_2)</td>
<td>0.08</td>
<td>0</td>
<td>−0.11</td>
<td>0.08</td>
</tr>
<tr>
<td>(X_3)</td>
<td>0.03</td>
<td>0.06</td>
<td>0</td>
<td>−0.03</td>
</tr>
<tr>
<td>(X_4)</td>
<td>0.01</td>
<td>−0.02</td>
<td>0.01</td>
<td>0</td>
</tr>
<tr>
<td>(X_5)</td>
<td>0</td>
<td>−0.06</td>
<td>0.01</td>
<td>−0.11</td>
</tr>
</tbody>
</table>

Total difference between relations matrix for the reference map and relations matrix for the learned maps was calculated according to the equation:

\[
J_r = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} (r_{j,i} - r_{j,i}^*)^2},
\]

where \(r_{j,i}^*\) is the reference value of the relation \(r_{j,i}\).

Total difference \(J_r = 0.63\) was obtained as a result of one-step algorithm \((m_1 = 0, m_2 = 0)\). Increasing the number of steps \(m_1\) led to reduced this value. Total difference \(J_r = 0.55\) was obtained as result of multi-step method \((m_1 = 2, m_2 = 0)\).

Figures 4 and 5 illustrate the results of the learning of CM for the number of steps \(m_1 = 0\) \((\alpha_0 = 1, \beta_0 = 30, \lambda_0 = 100)\) and \(m_1 = 2\) \((\alpha_0 = 0.6, \alpha_1 = 0.3, \alpha_2 = 0.1, \beta_0 = 30, \lambda_0 = 100)\). Figure 6 shows the results of testing the operation of learned maps.

Obtained results show that increasing the number of steps \(m_1\) gives certain possibilities to get quicker values of target CM weights and consequent improvement the operation of the learned CM.
5.2. Stability of the learned map. In order to analyze the influence of the steps number \((m_1, m_2)\) on the stability of the system, cognitive map was initialized and learned on the basis of real numerical data taken from the UCI Machine Learning Repository [19], and then used to the valuation of housing. The cognitive map with the following concepts was analyzed:

- \(X_1\) – CRIM: per capita crime rate by town,
- \(X_2\) – ZN: proportion of residential land zoned for lots over 25,000 sq.ft.,
- \(X_3\) – INDUS: proportion of non-retail business acres per town,
- \(X_4\) – CHAS: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise),
- \(X_5\) – NOX: nitric oxides concentration (parts per 10 million),
- \(X_6\) – RM: average number of rooms per dwelling,
- \(X_7\) – AGE: proportion of owner-occupied units built prior to 1940,
- \(X_8\) – DIS: weighted distances to five Boston employment centers,
- \(X_9\) – RAD: index of accessibility to radial highways,
- \(X_{10}\) – TAX: full-value property-tax rate per $10,000,
- \(X_{11}\) – PTRATIO: pupil-teacher ratio by town,
- \(X_{12}\) – B: 1000 \((Bk – 0.63)\) \(^2\) where \(Bk\) is the proportion of blacks by town,
- \(X_{13}\) – LSTAT: % lower status of the population,
- \(X_{14}\) – MEDV: Median value of owner-occupied homes in $1000’s – output of the system.

The structure of initialized map is presented in Fig. 7.

The map was learned based on 450 records using gradient method type (3), (5). Another 56 records were used in testing the operation of learned maps. Below the part of relations matrix for the map learned using one-step gradient method (Table 4) and the multi-step gradient method (Table 5) are shown.

Final values of the relations matrixes differ significantly, which has an influence on the operation of learned maps. Figure 8 shows the results of the learning of CM for the number of steps \(m_1 = 0\) \((\alpha_0 = 1, \beta_0 = 10, \lambda_0 = 10)\) and \(m_1 = 3\) \((\alpha_0 = 0.4, \alpha_1 = 0.3, \alpha_2 = 0.2, \alpha_3 = 0.1, \beta_0 = 10, \lambda_0 = 10)\).
Analysis of multi-step algorithms for cognitive maps learning

Figure 9 illustrates the results of testing the map learned with the number of steps \( m_1 = 0 \) (\( \alpha_0 = 1, \beta_0 = 10, \lambda_0 = 10 \)) and \( m_1 = 3 \) (\( \alpha_0 = 0.4, \alpha_1 = 0.3, \alpha_2 = 0.2, \alpha_3 = 0.1, \beta_0 = 10, \lambda_0 = 10 \)). Map learned with the multi-step method reached the stable state quicker than map learned with the one-step algorithm.

\[
J_P = \frac{1}{n_T} \sum_{t=n_L+1}^{n_T} \left( \frac{1}{n} \sum_{i=1}^{n} (X_i(t) - Z_i(t))^2 \right) \cdot 100\%, \quad (10)
\]

where \( n_T \) – the number of the test records, \( n_L \) – the number of the learning records.

5.3. Average percentage prediction error. In order to analyze the influence of the steps number \( (m_1, m_2) \) on the prediction error, a cognitive map was initialized and learned on the basis of real numerical and symbolic data taken from the UCI Machine Learning Repository [20, 21], and then used to the diagnosis of hepatitis. Optimal in some sense parameters of learning were chosen based on minimization of an average percentage prediction error, described as follows [3]:

The cognitive map with the following concepts was analyzed:

- \( X_1 \) – Class: DIE, LIVE,
- \( X_2 \) – AGE: 10, 20, 30, 40, 50, 60, 70, 80,
- \( X_3 \) – SEX: male, female,
- \( X_4 \) – STEROID: no, yes,
- \( X_5 \) – ANTIVIRALS: no, yes,
- \( X_6 \) – FATIGUE: no, yes,
- \( X_7 \) – MALAISE: no, yes,
- \( X_8 \) – ANOREXIA: no, yes,
- \( X_9 \) – LIVER BIG: no, yes,
- \( X_{10} \) – LIVER FIRM: no, yes,
- \( X_{11} \) – SPLEEN PALPABLE: no, yes,
- \( X_{12} \) – SPIDERS: no, yes,
- \( X_{13} \) – ASCITES: no, yes,
- \( X_{14} \) – VARICES: no, yes,
- \( X_{15} \) – BILIRUBIN: 0.39, 0.80, 1.20, 2.00, 3.00, 4.00,
- \( X_{16} \) – ALK PHOSPHATE: 33, 80, 120, 160, 200, 250,
- \( X_{17} \) – SGOT: 13, 100, 200, 300, 400, 500,
- \( X_{18} \) – ALBUMIN: 2.1, 3.0, 3.8, 4.5, 5.0, 6.0,
- \( X_{19} \) – PROTIME: 10, 20, 30, 40, 50, 60, 70, 80, 90,
- \( X_{20} \) – HISTOLOGY: no, yes.

The structure of the initialized map is presented in Fig. 11.
The map was learned for various parameters based on 141 records using the gradient method type (3), (5) and then, using the NHL algorithm type (3), (8). Another 14 records were used in testing the operation of learned maps. Below the part of relations matrix for the map learned using one-step gradient method (Table 6) and multi-step gradient method (Table 7) are shown.

**Table 6**

Relations matrix of the map learned with the parameters: \( m_1 = 0, m_2 = 0, \alpha_0 = 1, \beta_0 = 50, \lambda_0 = 100 \) (one-step method)

<table>
<thead>
<tr>
<th>( X_4 )</th>
<th>( X_5 )</th>
<th>( X_6 )</th>
<th>( X_7 )</th>
<th>( X_8 )</th>
<th>( X_9 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( X_4 )</td>
<td>0</td>
<td>-0.08</td>
<td>0.02</td>
<td>-0.01</td>
<td>-0.06</td>
</tr>
<tr>
<td>( X_5 )</td>
<td>0.03</td>
<td>0</td>
<td>0.12</td>
<td>0.09</td>
<td>0.12</td>
</tr>
<tr>
<td>( X_6 )</td>
<td>0</td>
<td>0.09</td>
<td>0</td>
<td>-0.11</td>
<td>0.21</td>
</tr>
<tr>
<td>( X_7 )</td>
<td>0.23</td>
<td>0.23</td>
<td>-0.01</td>
<td>0</td>
<td>0.21</td>
</tr>
<tr>
<td>( X_8 )</td>
<td>-0.07</td>
<td>0.25</td>
<td>0.01</td>
<td>0.06</td>
<td>0</td>
</tr>
<tr>
<td>( X_9 )</td>
<td>0.11</td>
<td>-0.03</td>
<td>-0.16</td>
<td>0.21</td>
<td>0.12</td>
</tr>
<tr>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
</tr>
</tbody>
</table>

**Table 7**

Relations matrix of the map learned with the parameters: \( m_1 = 5, m_2 = 5, \alpha_0 = 0.4, \alpha_1 = 0.2, \alpha_2 = 0.1, \alpha_3 = 0.1, \alpha_4 = 0.1, \alpha_5 = 0.1, \beta_0 = 10, \beta_1 = 10, \beta_2 = 10, \beta_3 = 10, \beta_4 = 5, \beta_5 = 5 \), \( \lambda_0 = \lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = \lambda_5 = 100 \) (multi-step method)

<table>
<thead>
<tr>
<th>( X_4 )</th>
<th>( X_5 )</th>
<th>( X_6 )</th>
<th>( X_7 )</th>
<th>( X_8 )</th>
<th>( X_9 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( X_4 )</td>
<td>0</td>
<td>0.07</td>
<td>0</td>
<td>0.02</td>
<td>0.06</td>
</tr>
<tr>
<td>( X_5 )</td>
<td>0.05</td>
<td>0</td>
<td>0.03</td>
<td>0.06</td>
<td>0.11</td>
</tr>
<tr>
<td>( X_6 )</td>
<td>0.03</td>
<td>0.09</td>
<td>0</td>
<td>0.01</td>
<td>0.13</td>
</tr>
<tr>
<td>( X_7 )</td>
<td>0.1</td>
<td>0.15</td>
<td>0</td>
<td>0</td>
<td>0.13</td>
</tr>
<tr>
<td>( X_8 )</td>
<td>0.02</td>
<td>0.16</td>
<td>0</td>
<td>0.04</td>
<td>0</td>
</tr>
<tr>
<td>( X_9 )</td>
<td>0.06</td>
<td>0.09</td>
<td>-0.04</td>
<td>0.08</td>
<td>0.11</td>
</tr>
<tr>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
</tr>
</tbody>
</table>

Figure 12 shows the results of the learning of CM for the number of steps \( m_1 = 0, m_2 = 0 (\alpha_0 = 1, \beta_0 = 50, \lambda_0 = 100) \) and \( m_1 = 5, m_2 = 5 (\alpha_0 = 0.4, \alpha_1 = 0.2, \alpha_2 = 0.1, \alpha_3 = 0.1, \alpha_4 = 0.1, \alpha_5 = 0.1, \beta_0 = 10, \beta_1 = 10, \beta_2 = 10, \beta_3 = 10, \beta_4 = 5, \beta_5 = 5, \lambda_0 = \lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = \lambda_5 = 100) \).

![Fig. 12. Obtained values \( X_{16}(t) \) and the desired values \( Z_{16}(t) \) during learning with using gradient method type (3), (5)](image)

Concepts values during learning with one-step method and multi-step method are similar. However, final values of the relations matrices are significantly different, which has influence on the operation of learned maps. Figure 13 presents the results of testing the operation of learned maps.

![Fig. 13. Obtained values \( X_{16}(t) \) and the desired values \( Z_{16}(t) \) during testing](image)

Table 8 presents selected results of the impact of steps number and learning parameters on the prediction error (\( J_P \)) for a map learned with the gradient method.

**Table 8**

Chosen results of analysis of the multi-step supervised algorithm based on gradient method (\( \lambda_1 = 100, i = 0, 1, \ldots, 5 \))

<table>
<thead>
<tr>
<th>( m_1 )</th>
<th>( m_2 )</th>
<th>( \alpha_0 )</th>
<th>( \alpha_1 )</th>
<th>( \alpha_2 )</th>
<th>( \alpha_3 )</th>
<th>( \alpha_4 )</th>
<th>( \alpha_5 )</th>
<th>( \beta_0 )</th>
<th>( \beta_1 )</th>
<th>( \beta_2 )</th>
<th>( \beta_3 )</th>
<th>( \beta_4 )</th>
<th>( \beta_5 )</th>
<th>( J_P[%] )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7.388</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7.24</td>
</tr>
<tr>
<td>0</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>7.186</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0.6</td>
<td>0.3</td>
<td>0.1</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7.26</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0.4</td>
<td>0.2</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>7.211</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0.6</td>
<td>0.3</td>
<td>0.1</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7.083</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>0.4</td>
<td>0.2</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>6.991</td>
</tr>
</tbody>
</table>

Prediction error \( J_P = 7.388\% \) was obtained as a result of one-step algorithm (\( m_1 = 0, m_2 = 0 \)). Increasing the number of steps of the gradient method led to reduced average percentage prediction error. The error minimum (\( J_P = 6.991\% \)) was obtained for the steps number: \( m_1 = 5, m_2 = 5 \).

Table 9 presents chosen results of the impact of steps number and learning parameters on the prediction error (\( J_P \)) for map learned with NHL algorithm.

**Table 9**

Chosen results of analysis of the multi-step supervised algorithm based on NHL algorithm (\( \lambda_1 = 100, i = 0, 1, \ldots, 2 \))

<table>
<thead>
<tr>
<th>( m_1 )</th>
<th>( m_2 )</th>
<th>( \alpha_0 )</th>
<th>( \alpha_1 )</th>
<th>( \alpha_2 )</th>
<th>( \beta_0 )</th>
<th>( \beta_1 )</th>
<th>( \beta_2 )</th>
<th>( J_P[%] )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.03</td>
<td>0</td>
<td>0</td>
<td>7.002</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.02</td>
<td>0.01</td>
<td>0</td>
<td>7.001</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.01</td>
<td>0.01</td>
<td>0</td>
<td>7</td>
</tr>
</tbody>
</table>

The prediction error \( J_P = 7.002\% \) was obtained as a result of one-step algorithm (\( m_1 = 0, m_2 = 0 \)). Increasing the
number of steps $m_2$ led to reduced average percentage prediction error. The error minimum ($J_P = 7\%$) was obtained for the steps number: $m_1 = 0$, $m_2 = 2$.

6. Conclusions

The paper contains the description of cognitive maps and multi-step algorithms for CM learning. A simulation analysis of the algorithm functioning was realized based on ISEMK software tool. Learning of CM was based on a reference map and real data. Chosen results of the analysis, showing the influence of the implementation of the multi-step technique on the speed of convergence of the learning algorithm and the system functioning, were presented. It can be stated that the implementation of the multi-step technique gives certain possibilities to get quicker values of target CM weights. The advantage of the application of multi-step algorithms is also the improvement of the functioning of the learned system by reduction of average percentage prediction error and obtaining better stabilization.

There are plans of further development of multi-step algorithms as possible generalization of known one-step methods of CM learning.

REFERENCES


