Journal of Signal Processing, Vol.27, No.4, pp.69-73, July 2023

SELECTED PAPER AT NCSP’23

Multilayered Cortical Learning Algorithm for Forecasting Time-Series Data with Probabilistically Changing Trends

Kazushi Fujino[1], Takeru Aoki[2], Keiki Takadama[1] and Hiroyuki Sato[1]

[1]Graduate School of Informatics and Engineering, The University of Electro-Communications
1-5-1 Chofugaoka, Chofu, Tokyo 182-8585, Japan
E-mail: k.fujino@mail.uec.jp, keiki@inf.uec.ac.jp, h.sato@uec.ac.jp
[2]Faculty of Engineering, Department of Information and Computer Technology, Tokyo University of Science
1-3 Kagurazaka, Shinjuku-ku, Tokyo 162-8601, Japan
E-mail: aoki@rs.tus.ac.jp

Abstract

In this paper, a burst-based multilayered cortical learning algorithm (BM-CLA) for forecasting trend-changing time-series data is proposed. CLA predicts time-series data while adjusting synapse relationships online. However, the forecast accuracy of the conventional CLA deteriorates with trend-changing time-series data, in which several time-series trends are switched over time. The proposed BM-CLA detects trend changes based on multilayered CLA predictors. Experimental results using multiple artificial time-series data with probabilistically changing trends showed that BM-CLA achieves results that are better or comparable to those of conventional CLAs with different specifications and the long short-term memory (LSTM), which is a neural network-based forecast algorithm.

1. Introduction

The cortical learning algorithm (CLA) is a time-series forecast method. CLA is designed on the basis of the hierarchical temporal memory (HTM) [1], which is a model of the human neocortex. The CLA predictor consists of *columns* and *cells* as memory components and *synapses* as linking components. Each of the columns and cells transits among multiple states. A state combination of columns represents an input value at each time step and a state combination of cells represents an input value in the time-series data. CLA forecasts time-series data by changing the states of columns and cells on the basis of synapse relationships. It was reported that CLA achieved a higher forecast accuracy than the long short-term memory (LSTM) [2], a neural network-based time-series forecast method, in real-world forecast tasks of electricity load [3] and taxi demand [4]. However, the forecast accuracy of the conventional CLA deteriorates when the time-series trend changes, even if the next incoming trend was already received previously. This is caused by the conventional CLA that updates synapse relationships to fit only to the recent time-series data.

To improve the forecast accuracy of CLA with time-series data with changing trends, in this work, we propose a burst-based multilayered CLA (BM-CLA). The proposed BM-CLA has two CLA predictors: the lower and upper layers. The lower layer is the conventional predictor but warns burst columns of a prediction failure to the upper layer, which is a new predictor. The upper layer recognizes the pattern of burst columns in the lower layer and encourages the state change of cells in the lower layer in accordance with the current time-series trend. To verify the effectiveness of the proposed BM-CLA, we compare the performance characteristics of the conventional LSTM, CLAs with different specifications, and the proposed BM-CLA in forecasting time-series data with probabilistically changing trends.

This paper is based on [5] and for the invited special issue. This paper additionally involves search for parameters of the proposed BM-CLA.

2. Conventional Cortical Learning Algorithm

2.1 Predictor

Figure 1 shows the CLA predictor [1]. Every time step

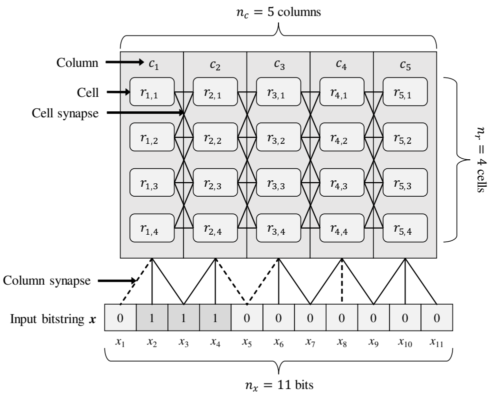

Figure 1: CLA predictor [1]

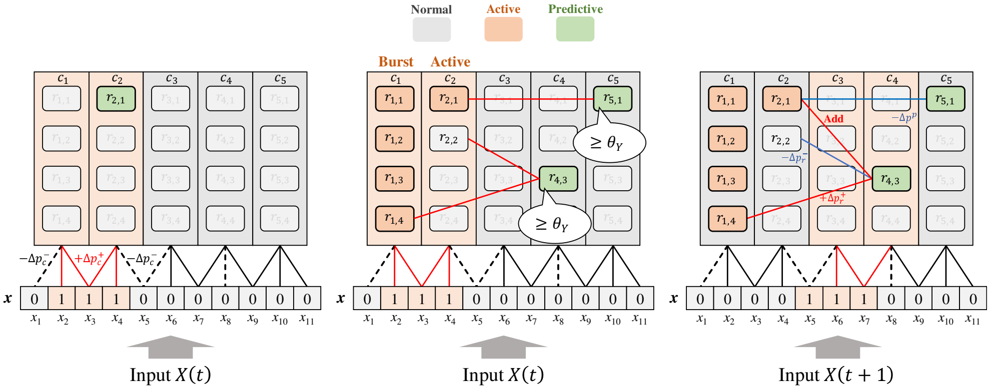

Figure 2: Prediction process of conventional CLA

t, it receives an input bit string $\boldsymbol{x} \in \{0,1\}^{n_x}$ converted from input value $X(t)$. The predictor has n_c columns c_i $(i = 1,2,\ldots,n_c)$. Each column c_i has n_r cells $r_{i,j}$ $(j = 1,2,\ldots,n_r)$. Each column has two states: *normal* and *active*. A set of active columns represents the input $X(t)$ at the time step t. Each cell has three states: *normal*, *active*, and *predictive*. A set of active cells represents the input $X(t)$ in the current context. A set of predictive cells represents the input value $X(t+1)$ at the next time step $t+1$ in the context. Each synapse has a permanence value p. For a given connection threshold θ_p, the synapse is connected if $p \geq \theta_p$ and disconnected otherwise. Figure 1 shows connected synapses as solid lines and disconnected ones as dashed lines.

2.2 Procedure

Figure 2 shows the prediction process of CLA. In each time step t, CLA converts the input value $X(t)$ to an input bit string $\boldsymbol{x}$. In this work, we use the chunk converter [3]. CLA sorts columns in descending order of the number of synapses connected to 1 on the input bit string $\boldsymbol{x}$ and makes the top A_c columns the active state. The left diagram in Fig. 2 shows that $A_c = 2$ columns c_1 and c_2 transit to the active state. For each active column, CLA increases the permanence p of the column synapse associated with a bit of 1 by Δp_c^+ and decreases the p of the column synapse associated with a bit of 0 by Δp_c^-. Thus, column synapses work to construct relations between input data bits and columns.

CLA makes at least one cell $r_{i,j}$ the active state in each active column c_i on the basis of the following two conditions. First, predictive cells $r_{i,j}$ in each active column c_i transit to the active state. Second, all cells $r_{i,j}$ $(j = 1,\ldots,n_r)$ in each active column c_i transit to the active state if c_i does not include any predictive cells. The second condition is called *burst*, indicating a prediction error that the active column could not be predicted. In the left and middle of Fig. 2, we see that cell $r_{2,1}$ transits to the active state from the predictive state since column c_2 is in the active state. It is a prediction success. On the other hand, column c_1 bursts and all cells $r_{1,1}, r_{1,2}, \ldots, r_{1,4}$ transit to the active state since column c_1 is in the active state but does not have any predictive cells. It is a prediction failure.

CLA then gives the predictive state to cells with more than θ_Y connected cell synapses associated with active cells. The center diagram in Fig. 2 shows that cells $r_{4,3}$ and $r_{5,1}$ transit to the predictive state with threshold $\theta_Y = 1$.

For each cell transited from the predictive state to the active state, CLA increases the p of the cell synapse associated with active cells at the previous time step $t-1$ by Δp_r^+ and decreases the p of the cell synapse associated with normal cells at the previous time step $t-1$ by Δp_r^-. Also, CLA adds new cell synapses between cells transited from the predictive state to the active state and the active cell at the previous time step $t-1$. Also, for each cell transited from the predictive state to the normal state, CLA decreases the p of the cell synapse associated with active cells at the previous time step $t-1$ by Δp^p. In this way, CLA constructs cell synapse relations to make successful predictions. The right diagram in Fig. 2 shows that the predictive cell $r_{4,3}$ is in the active column c_4 and the predictive cell $r_{5,1}$ is in the normal column c_5. For the prediction success cell $r_{4,3}$, the p of the cell synapse from active $r_{1,4}$ is increased, the p of one from the normal $r_{2,2}$ is decreased, and a new synapse to active $r_{2,1}$ is added. For the prediction failure cell $r_{5,1}$, the p of the cell synapse from active $r_{2,1}$ is decreased. Finally, CLA outputs the predicted value $\tilde{X}(t+1)$ from the set of cells in the predictive state.

In this way, recent input values especially affect the latest synapse relationships and the synapse relationships for other time-series trends trained previously are gradually weakened. As a result, the conventional CLA deteriorates the prediction accuracy when the previous time-series trend is received again.

3. Proposal: Burst-Based Multilayered CLA (BM-CLA)

3.1 Predictor

Figure 3 shows the BM-CLA predictor, which has two CLA predictors called the upper and lower layers. BM-CLA

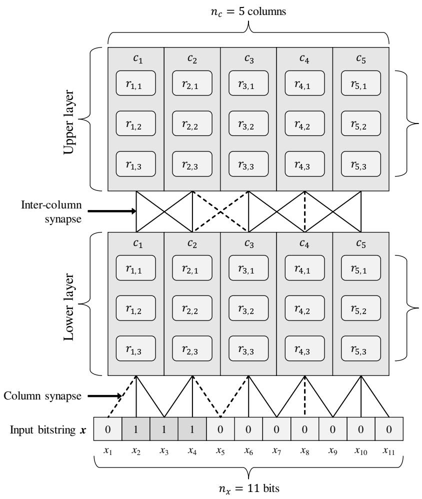

Figure 3: BM-CLA predictor

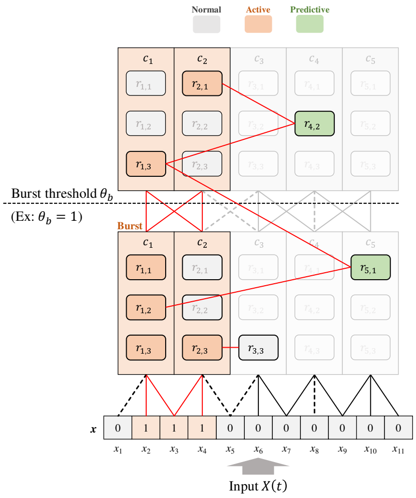

Figure 4: Prediction process of BM-CLA

sets intercolumn synapses between two layers.

3.2 Procedure

Figure 4 shows the prediction process of BM-CLA. In each time step t, the lower layer receives an input bit string $\boldsymbol{x}$. The lower layer makes A_c columns the active state and cells in the active columns the active state in the same manner as the conventional CLA. BM-CLA then counts the number of burst columns, n_b, in the lower layer. Figure 4 also shows an example where only column c_1 in the lower layer bursts and $n_b = 1$. For a user-defined threshold θ_b of the proposed BM-CLA, the lower layer informs the active column pattern to the upper layer when $n_b \geq \theta_b$. The lower layer does not inform anything otherwise and the upper layer maintains the column states. In Fig. 4, the threshold $\theta_b = 1$ is assumed. This example satisfies $n_b \geq \theta_b$ and shows that the lower layer informs the active column pattern of c_1 and c_2 to the upper layer.

In the upper layer, BM-CLA makes A_c columns the active state and cells in their active columns the active state in the same manner as the conventional CLA. Figure 4 shows that the upper layer makes columns c_1 and c_2 the active state and cells $r_{1,3}$ and $r_{2,1}$ the active state.

In both layers, BM-CLA makes cells with more than θ_Y cell synapses connected to active cells the predictive state. In BM-CLA, cells in the upper layer have cell synapses within the upper layer and cells in the lower layer have cell synapses from both the upper and lower layers. That is, the state transitions of cells in the lower layer are affected by the cell states in the upper layer.

In the upper layer of Fig. 4, we see that cell $r_{4,2}$ transits to the predictive state by $\theta_Y = 2$ cell synapses associated with two active cells $r_{1,3}$ and $r_{2,1}$ in the same upper layer. In the lower layer of Fig. 4, we see that cell $r_{5,1}$ transits to the predictive state by $\theta_Y = 2$ cell synapses associated with one active cell $r_{1,2}$ in the same lower layer and another one $r_{1,3}$ in a different upper layer.

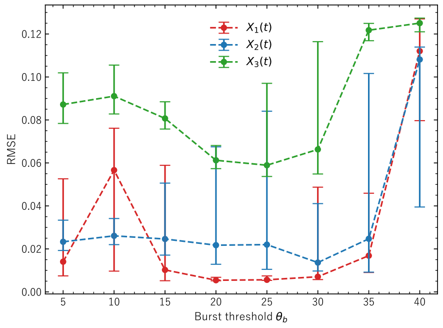

Figure 5: RMSE obtained by varying burst threshold θ_b

BM-CLA updates cell synapses in the same manner as the conventional CLA. Regarding the new cell synapse arrangement, the upper layer adds new cell synapses only in the same upper layer. The lower layer adds new cell synapses in both the upper and lower layers.

Finally, BM-CLA outputs the predicted value $\tilde{X}(t+1)$ from the set of cells in the predictive state in the lower layer.

In this work, we also employ the adaptive synapse adjustment (ASA) methodology [6], which adaptively controls the number of cell synapse generations on the basis of the partial prediction accuracy of a set of synapses. The adjustment strength is controlled by the parameter $w = 0.2$.

In this way, the upper layer keeps the same active cells and repeatedly informs them to the lower layer while the number of burst columns n_b in the lower layer is less than the burst threshold θ_b. This contributes to easily making specific cells in the lower layer into the predictive state. A large number of bursts in the lower layer indicates that BM-CLA receives an input value $X(t)$ that is significantly different from the prediction value $\tilde{X}(t)$. BM-CLA handles the number of burst columns, n_b, greater than or equal to θ_b as a trend change of the time-series data. BM-CLA then conducts a state change

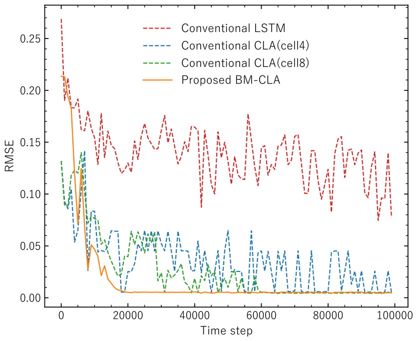

Figure 6: RMSE transition for $X_1(t)$

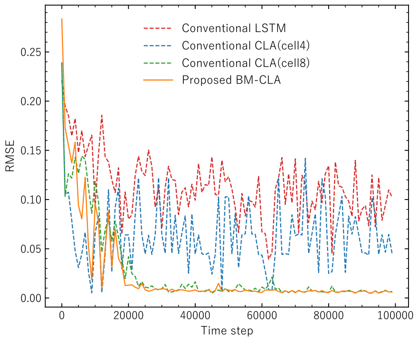

Figure 7: RMSE transition for $X_2(t)$

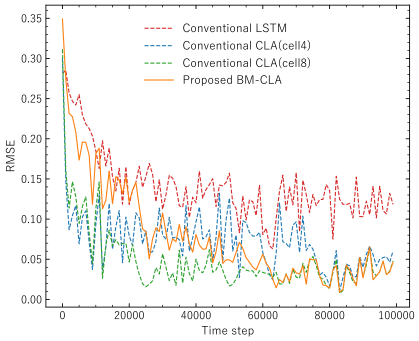

Figure 8: RMSE transition for $X_3(t)$

in the upper layer and easily makes the other cells in the lower layer into the predictive state. This contributes to making appropriate cells in the lower layer into the predictive state even if cell synapse relationships for other time-series trends received previously have been weakened. This prevents the deterioration of prediction accuracy when previously received time-series data are again received.

4. Experimental Settings

We used three trend-changing artificial time-series data $X_1(t)$, $X_2(t)$, and $X_3(t)$. $X_1(t)$ has two trends, the sine and the sawtooth waves. $X_2(t)$ has three trends, the sine, the sawtooth, and the triangular waves. $X_3(t)$ has three trends, which are the sine, the sawtooth, and the logistic map with $\alpha = 3.875$ waves. The sine, the sawtooth, and the triangular waves are cyclic and take up 100-time steps in one cycle. Each of $X_1(t)$, $X_2(t)$, and $X_3(t)$ randomly changes its trend every 100 time steps. The total number of time steps is 10^5.

We compared the conventional LSTM, the two conventional CLAs using $n_r = \{4, 8\}$ cells, and the proposed BM-CLA using $n_r = 4$ cells in each layer. For CLAs, the number of columns is set to $n_c = 2,048$. The length of the input bit string is set to $n_x = 421$. The number of active columns is set to $A_c = 40$. The activate threshold is set to $\theta_Y = 15$. The connected threshold of column synapses is set to $\theta_p = 0.1$. The amounts of increase and decrease in connection strength at column synapses are set to $\Delta p_c^+ = 0.1$ and $\Delta p_c^- = 0.00025225$, respectively. The connected threshold of cell synapses is set to $\theta_p = 0.21$. The amounts of increase and decrease in connection strength at cell synapses are set to $\Delta p_r^+ = 0.1$, $\Delta p_r^- = 0.1$, and $\Delta p^p = 0.05$. For LSTM, we used the window size 100 and Adam with a learning rate of 0.01 for optimization. Other settings are derived from [3]. We ran each algorithm 31 times for each of $X_1(t)$, $X_2(t)$, and $X_3(t)$. As a metric, we employed the root mean squared error (RMSE). The smaller the RMSE, the higher the forecast accuracy.

5. Experimental Results and Discussion

Figure 5 shows average RMSE values of the proposed BM-CLA in the last 50,000 time steps when the burst threshold θ_b varied. These results show that too small or large θ_b values deteriorate RMSE values. On the basis of the results, we use $\theta_b = 20$ for the proposed BM-CLA in the following experiments.

Figures 6–8 show median RMSE values over time steps. Figure 6 shows results for $X_1(t)$. We see that the proposed BM-CLA achieves the best performance from early time steps. Furthermore, the proposed BM-CLA is more stable than the other CLAs. Figure 7 shows results for $X_2(t)$. We see that the proposed BM-CLA also achieves the highest performance from early time steps and stable transition. Figure 8 shows results for $X_3(t)$. We see that the conventional CLA with 8 cells shows the smallest RMSE in the early stage. However, in the final stage, the proposed BM-CLA shows RMSE comparable to that of the conventional CLA with 8 cells.

6. Conclusions

To improve the prediction accuracy of CLA on time-series data with probabilistically changing trends, we proposed a burst-based multilayer CLA (BM-CLA). Experimental results using three time-series data with probabilistically changing trends showed that the proposed BM-CLA achieved results that are better or comparable to those of conventional CLAs with different specifications and LSTM.

In future work, we will study an adaptive adjustment of intercolumn synapses between the upper and lower layers.

References

[1] J. Hawkins, A. Subutai and D. Dubinsky: Hierarchical temporal memory including HTM cortical learning algorithms, Technical Report, Numenta Inc., 2010.

[2] S. Hochreiter and J. Schmidhuber: Long short-term memory, Neural Computation, Vol. 9, No. 8, pp. 1735–1780, 1997.

[3] T. Aoki, K. Takadama and H. Sato: Adaptive synapse arrangement in cortical learning algorithm, JACIII, Vol. 25, No. 4, pp. 450–466, 2021.

[4] Y. Cui, S. Ahmad and J. Hawkins: Continuous online sequence learning with an unsupervised neural network model, Neural Computation, Vol. 28, No. 11, pp. 2474–2504, 2016.

[5] K. Fujino, T. Aoki, K. Takadama and H. Sato: Multi-layer cortical learning algorithm for trend changing time-series forecast, 2023 RISP International Workshop on Nonlinear Circuits, Communications and Signal Processing (NCSP2023), pp. 281–284, 2023.

[6] K. Fujino, T. Aoki, K. Takadama and H. Sato: Adaptive synapse adjustment for multivariate cortical learning algorithm, Joint 12th International Conference on Soft Computing and Intelligent Systems and 21th International Symposium on Advanced Intelligent Systems (SCIS & ISIS2022), pp. 1–8, 2022.