Double-Layered Cortical Learning Algorithm for Time-Series Prediction

Takeru Aoki(✉), Keiki Takadama, and Hiroyuki Sato

The University of Electro-Communications, 1-5-1 Chofugaoka, Chofu, Tokyo 182-8585, Japan
{takeru-aoki,h.sato}@uec.ac.jp, keiki@inf.uec.ac.jp

Abstract. This work proposes a double-layered cortical learning algorithm. The cortical learning algorithm is a time-series prediction methodology inspired from the human neuro-cortex. The human neuro-cortex has a multi-layer structure, while the conventional cortical learning algorithm has a single layer structure. This work introduces a double-layered structure into the cortical learning algorithm. The first layer represents the input data and its context every time-step. The input data context presentation in the first layer is transferred to the second layer, and it is represented in the second layer as an abstract representation. Also, the abstract prediction in the second layer is reflected to the first layer to modify and enhance the prediction in the first layer. The experimental results show that the proposed double-layered cortical learning algorithm achieves higher prediction accuracy than the conventional single-layered cortical learning algorithms and the recurrent neural networks with the long short-term memory on several artificial time-series data.

Keywords: Cortical learning algorithm · Hierarchical temporal memory · Time-series data prediction

1 Introduction

Cortical learning algorithm (CLA) is a computational methodology based on hierarchical temporal memory [1–3], which is inspired by the human neuro-cortex. CLA is particularly suited to predict time-series data stream [4–6]. As related works on time-series prediction, the recurrent neural network (RNN) [7,8] with the long short-term memory (LSTM) [9] has been known as a representative and effective approach. It has been reported that CLA achieved higher prediction accuracy than LSTM on the taxi demand prediction task [10], and CLA is one of the promising time-series prediction algorithms.

The entire CLA predictor is called the *region*. The region is composed of multiple *columns*, and each column is composed of multiple *cells*. Each column has

T. Nakano (Ed.): BICT 2021, LNICST 403, pp. 33–44, 2021.
https://doi.org/10.1007/978-3-030-92163-7_4

column-synapses associating with input data bits. CLA uses column-synapses and encodes the input data value into internal data representation every time-step by making a subset of columns the *active* state. Also, CLA represents the input data context by making a subset of cells the *active* state and the input data incoming next time-step by making a subset of cells the *predictive* state. The conventional CLA predictor is composed of a single layer of one region. On the other hand, it has been known that the cerebral neocortex is composed of multiple layers. Since the multi-layering enhances the abstraction of internal data representation, we can expect the improvement of the prediction accuracy and the speeding up of the learning process in CLA.

In this work, we propose a double-layered CLA. Each layer is the conventional CLA predictor, and we propose interactions between layers. The first layer receives input data and transfers its data context representation to the second layer. The second layer makes a prediction based on the data context representation received from the first layer and makes feedback to the first layer. The first layer enhances the prediction based on the feedback from the second layer. We use multiple artificial time-series data for the experiments to verify the prediction performance of the proposed double-layered CLA. We have conducted a preliminary attempt on multi-layer CLA that tried some prototypes on an artificial time-series prediction [11]. The proposed double-layered CLA in this paper does not emphasize synapse connections for the first layers' active cells predicted by the second layer for stability. Also, the proposed double-layered CLA in this paper relaxes the condition to make cells the predictive state in the first layer with a parameter α, and the appropriate α is found out experimentally. The effectiveness of the proposed method is verified on several artificial time-series prediction tasks while comparing with simple conventional CLA and ANN-based LSTMs.

2 Cortical Learning Algorithm

2.1 Predictor Structure

Figure 1 shows the conventional structure of CLA predictor [4–6]. CLA receives an input data bit-string $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ every time-step t. The entire CLA predictor is called the region, which is composed of n_c columns c_i $(i = 1, 2, \ldots, n_c)$. Figure 1 shows the case with $n_c = 5$ columns.

Each column c_i has two states: *normal* and *active*. Each column has n_{cy} column-synapses $c_i.y_k$ $(k = 1, 2, \ldots, n_{cy})$, which construct the relation with input data bits. Each column-synapse $c_i.y_k$ has permanence value $c_i.y_k.p$, which determines the connection or the disconnection of the synapse. Column-synapse $c_i.y_k$ is connected when $c_i.y_k.p \geq \theta_c$ and disconnected when $c_i.y_k.p < \theta_c$, where θ_c is the connection threshold of column-synapses.

Each column c_i involves n_r cells $r_{i,j}$ $(j = 1, 2, \ldots, n_r)$. Figure 1 shows the case with $n_r = 5$ cells. Each cell has three states: *normal*, *active*, and *predictive*. Also, each cell $r_{i,j}$ has cell-synapses $r_{i,j}.y_k$ $(k = 1, 2, \ldots)$, which construct relations with other cells. Each cell-synapse $r_{i,j}.y_k$ has permanence value $r_{i,j}.y_k.p$.

Cell-synapse $r_{i,j}.y_k$ is connected when $r_{i,j}.y_k.p \geq \theta_r$ and disconnected when $r_{i,j}.y_k.p < \theta_r$, where θ_r is the connection threshold of cell-synapses.

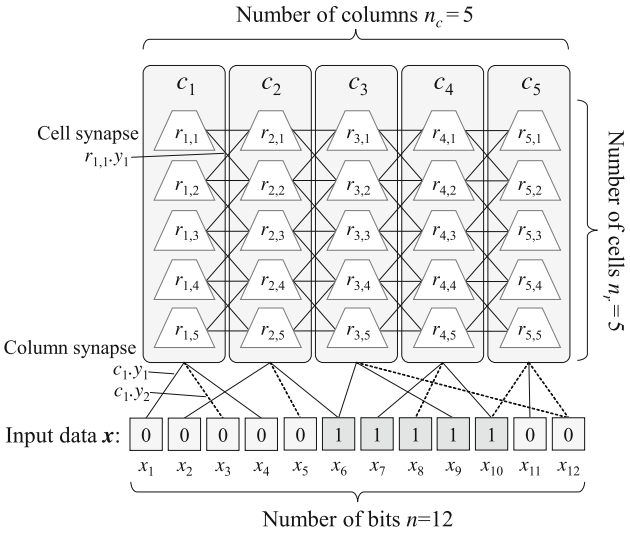

Fig. 1. CLA predictor

2.2 Algorithm

CLA repeats the following procedure every time-step t.

- Data encoding (Sect. 2.3)
- Spatial pooling (Sect. 2.4)
- Temporal pooling (Sect. 2.5)
- Data decoding (Sect. 2.6)

Details of each procedure are described as follows.

2.3 Data Encoding

CLA treats binary bit-string as input data value every time-step t. For a real value input data $X(t) \in [X^{\min}, X^{\max}]$ received at time-step t, CLA converts it to a binary bit-string $\boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in \{0, 1\}^n$ by using a chunk encoding method [6], which converts a real value into a bit-string involving continuous w chunk bits of 1. Figure 1 shows the case with the total bit length $n = 12$ and the chunk bit length $w = 5$.

2.4 Spatial Pooling

Spatial pooling is the process to convert the input data bit-string $\boldsymbol{x}$ into an internal representation in the CLA predictor. Specifically, CLA converts $\boldsymbol{x}$ into an active state combination of columns. CLA finds n_{ac} columns with many column-synapses connected to ones on the input data bit-string $\boldsymbol{x}$ and low active frequency and makes them the active state. CLA then updates the permanence value of column-synapses of the active columns to emphasize the internal data representation by the active pattern of columns. CLA increases the permanence value of column-synapses associated with one on the input data bit-string by p_c^+. CLA decreases the permanence value of column-synapses associated with zero on the input data bit-string by p_c^-.

2.5 Temporal Pooling

Temporal pooling represents input data context by an active state combination of cells and data incoming next time-step $t+1$ by a predictive state combination of cells. CLA first makes cells in the active columns the active state to represent the input data context. For each active column, CLA makes a predictive cell the active state if it exists. CLA makes all cells in the active column the active state otherwise. CLA then updates the permanence values of cell-synapses associated with the active cells transited from the predictive state to emphasize the cell-synapse network that succeeds the prediction. For each active cell, CLA increases the permanence value of cell-synapses associated with active cells at the previous time-step $t-1$ by p_r^+. Also, CLA decreases the permanence value of cell-synapses associated with non-active cells at the previous time-step $t-1$ by p_r^-.

Next, CLA makes cells with more than n_{ar} cell-synapses connected to the active cells at the current time-step t the predictive state. In this way, CLA internally represents the input data incoming next time-step $t+1$ as a combination of cells in the predictive state.

2.6 Data Decoding

CLA decodes the combination of cells in the predictive state to the data format same as the input data bit-string. In this work, we use the sparse distributed representations classifier (SDRC) [3] for the data decoding.

3 Proposal: Double-Layered Cortical Learning Algorithm

3.1 Predictor Structure

Figure 2 shows the predictor structure of the double-layered CLA we propose in this work. In the proposed double-layered CLA, the region of the conventional CLA shown in Fig. 1 is accumulated. The lower layer is called the first layer, and the higher layer is called the second layer. For the illustration, although cells in

the first layer are positioned side-by-side in this figure, the predictor structures in the first and second layers are the same.

The first and second layers are connected by interlayer-synapses, which are column-synapses in the second layer. Each interlayer-synapse associates a cell in the first layer with a column in the second layer. The interlayer-synapses transfer cell information from the first layer to the second layer. Conversely, the interlayer-synapses transfer predictive information from the second layer to the first layer. In the double-layered CLA, the role of the first layer is similar to the conventional CLA, and the second layer affects to cell states in the first layer.

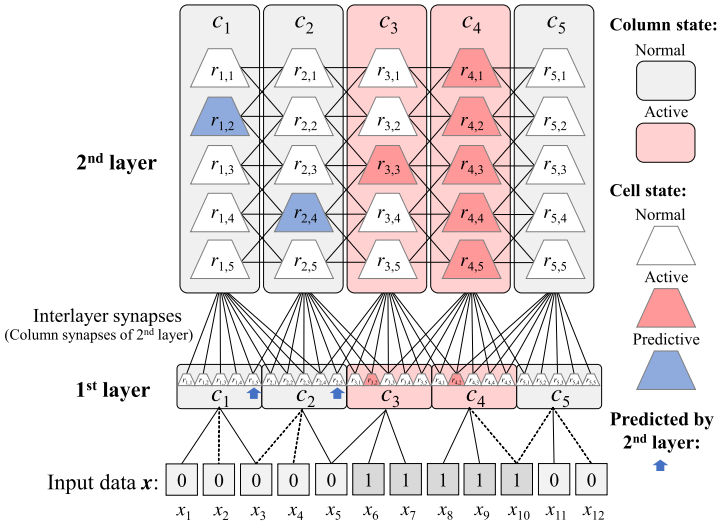

Fig. 2. Proposed double-layered CLA predictor

3.2 Algorithm

The first layer receives the input data $\boldsymbol{x}$ every time-step t. In the first layer, the spatial pooling makes a subset of columns the active state, the temporal pooling makes a subset of cells the active state. They are the same manner as the conventional CLA. Figure 2 show an example that columns c_3 and c_4 become the active state and represent the input data $\boldsymbol{x}$ in the first layer as the result of the spatial pooling. Also, cells $r_{3,2}$ and $r_{4,2}$ become the active state and represent the data context in the first layer as the result of the temporal pooling. Then, the first layer transfers the active cell pattern to the second layer through interlayer-synapses. In this process, the normal cell in the first layer is treated as 0, the active cell in the first layer is treated as 1, and the second layer performs the spatial pooling in the same manner as the conventional CLA. That is, the second layer makes a subset of columns the active state to represent the active cell pattern in the first layer. In the example of Fig. 2, we set 1 to cells

$r_{3,2}$ and $r_{4,2}$ and 0 to other cells in the first layer. The second layer receives them, and columns c_3 and c_4 become the active state in the second layer. Then, the second layer also performs the temporal pooling that makes a subset of cells in the second layer the active state and another subset of cells in the second layer the predictive state in the same manner as the conventional CLA. In the example of Fig. 2, cells of columns c_3 and c_4 in the second layer becomes the active state and makes cells $r_{1,2}$ and $r_{2,4}$ the predictive state in the second layer. Next, the second layer transfers the predictive cell information to the first layer through the interlayer-synapses, and the first layer determines predictive cells while considering the prediction from the second layer. This process is called the feedback in this work.

3.3 Feedback

The proposed double-layered CLA makes a subset of cells in the first layer the predictive state while considering the predictive information from the second layer. For the first layer's cells predicted from the second layer, the proposed double-layered CLA relaxes the condition to be the predictive state in the first layer. The example of Fig. 2 indicates that cells $r_{1,5}$ and $r_{2,5}$ in the first layer are predicted from the second layer. The proposed double-layered CLA relaxes the condition to be the predictive state for these cells. The proposed double-layered CLA uses two ways to relax the condition to make cells the predictive state in the first layer by employing a relaxing parameter $\alpha = [0, 1]$.

The first way to relax the condition to make cells the predictive state is the decrease of the connection threshold θ_r of cell-synapses. The proposed double-layered CLA employs the connection threshold 0 for cells predicted from the second layer such as $r_{1,5}$ and $r_{2,5}$ in the first layer of Fig. 2. For other cells not predicted from the second layer, the conventional connection threshold θ_r is used. Since each cell must have more than n_{ac} cell-synapses connected to active cells to be the predictive state, the decrease of the connection threshold to 0 contributes to increasing the possibility to be the predictive state by increasing the number of connected cell-synapses.

The second way to relax the condition to make cells the predictive state is the decrease of the threshold to be the predictive state more directly. That is, the proposed double-layered CLA reduces the number of cell-synapses n_{ar} connected to active cells to be the predictive state. The proposed double-layered CLA employs the number of cell-synapses $\alpha \cdot n_{ar}$ connected to active cells for cells predicted from the second layer such as $r_{1,5}$ and $r_{2,5}$ in the first layer of Fig. 2. For other cells not predicted from the second layer, the conventional n_{ar} is used. This helps to make cells predicted from the second layer the predictive state in the first layer even if the cells do not have more than n_{ar} connected cell-synapses to active cells.

$\alpha = 1$ indicates that any prediction information from the second layer is not reflected in the first layer and brings the same results as the conventional single-layered CLA. Influence of the second layer increases by decreasing the relaxing parameter α from 1.

3.4 Interlayer-Synapse Arrangement

Each interlayer-synapse, column-synapse of the second layer, associates a column in the second layer with a cell in the first layer. The proposed double-layered CLA dynamically arranges interlayer-synapses over time. Concretely, during the spatial pooling in the second layer, we arrange interlayer-synapses associating second-layer columns with a low active frequency and active cells in the first layer when the number of active columns in the second layer is less than n_{ac}.

The dynamic interlayer-synapse arrangement starts after 10^4 time-steps since it cannot construct a valuable network until the lower layer is unstable.

3.5 Expected Effects

In the proposed double-layered CLA, we can expect to improve the learning speed of CLA since the feedback improves the learning efficiency. Also, we can expect to improve the prediction accuracy since the feedback from the second layer modifies the prediction in the first layer even if the prediction in the first layer does not work well.

4 Experimental Settings

4.1 Algorithms

To verify the effectiveness of the proposed CLA, we compared the conventional CLA [6], the simple CLA [12], the proposed CLA, and three RNN-based LSTMs using Adam, RMSprop, and SGD, which are network optimization algorithms.

4.2 Benchmark Time-Series Data

As benchmark time-series data, we use sine $X_s(t)$ and combined sine $X_c(t, m)$ as cyclic time-series data. Also, we use logistic mapping based $X_l(t)$ as non-cyclic time-series data. These benchmark time-series data are defined as follows:

$$X_s(t) = \frac{1}{2} \cdot \sin\left(\frac{(t-1)\cdot\pi}{50}\right) + \frac{1}{2}, \tag{1}$$

$$X_c(t, m) = \frac{1}{2} \cdot \sum_{k=1}^{m} \frac{1}{2k-1} \cdot \sin\left(\frac{(t-1)\cdot(2k-1)\cdot\pi}{50}\right) + \frac{1}{2}, \tag{2}$$

$$X_l(t) = \begin{cases} 0.4, & \text{if } t = 1, \\ 3.6 \cdot X_l(t-1) \cdot \{1 - X_l(t-1)\}, & \text{otherwise.} \end{cases} \tag{3}$$

For combined sine $X_c(t, m)$, we use $m = \{2, 3, 4, 5\}$. The range of time-steps is set to $t \in [1, 10^5]$. Input value range is set to $[X^{\min}, X^{\max}] = [-0.01, 1.01]$.

Table 1. Parameters of CLA in experiment

Name	Value
Length of the data bit-string n	421
Chunk length w	21
Number of columns n_c	2048
Number of column-synapses n_{cy}	22
Connection threshold of the column-synapses θ_c	0.1
Number of active columns n_{ac}	40
Increase permanence value for column synapses p_c^+	0.05
Decrease permanence value for column synapses p_c^-	0.025225
Number of cells in each column n_r in the first layer	4
Number of cells in each column n_r in the second layer	5
Connection threshold of the cell-synapses θ_r	0.5
Number of connected active cells to be the predictive state n_{ar}	15
Increase permanence value for cell synapses p_r^+	0.1
Decrease permanence value for cell synapses p_r^-	0.1

4.3 Metric

As a metric to assess the prediction accuracy of the time-series data, we use the prediction error $e(t)$ given by

$$e(t) = \sum_{\tau=t-99}^{t} |\bar{X}(\tau+1) - X(\tau+1)|, \tag{4}$$

where, $\bar{X}(t+1)$ is the prediction value for the next time-step $t+1$, $X(t+1)$ is the true value at the next time-step $t+1$. Thus, $e(t)$ is the summation of the difference between the prediction and the true values every 100 time-steps. The smaller $e(t)$, the better prediction accuracy. In this work, we calculate $e(t)$ for $t \in \{100, 200, 300, 400, \ldots, 10^5\}$ in this work.

4.4 Parameters

As the implementation of the proposed CLAs, we employ the simple CLA, which involves the deterministic arrangement of the initial column-synapses, the fixed initial permanence value, and the aggregation of the column-synapse arrangement. For the three CLAs, the length of the data bit-string is set to $n = 421$ bits, and the chunk length is set to $w = 21$ bits. For the spatial pooling, the number of columns is set to $n_c = 2048$, the number of column-synapses is set to $n_{cy} = 22$. The connection threshold of the column-synapses is set to $\theta_c = 0.1$, the number of active columns at each time-step is set to $n_{ac} = 40$, the increase and the decrease permanence values are respectively $p_c^+ = 0.05$ and $p_c^- = 0.025225$.

For the temporal pooling, the number of cells in each column is set to $n_r = 4$ and 5 for the first and the second layers, respectively. The connection threshold of the cell-synapses is set to $\theta_r = 0.5$, the number of connected active cells to be the predictive state is set to $n_{ar} = 15$. Also, the increase and the decrease permanence values are respectively $p_r^+ = 0.1$ and $p_r^- = 0.1$. In the proposed double-layered CLA, we use the above parameters for both layers. Table 1 shows the parameters of CLA used in experiment.

We employ three LSTMs with different network optimization methods, Adam [13], RMSprop [14], and SGD. The Keras implementation [15] is employed in this paper. The neural network in the first layer is densely connected. The second layer is constructed as the LSTM with a tanh activation function. The output layer also has the densely connected neural network. For the loss function, the mean-squared error is used.

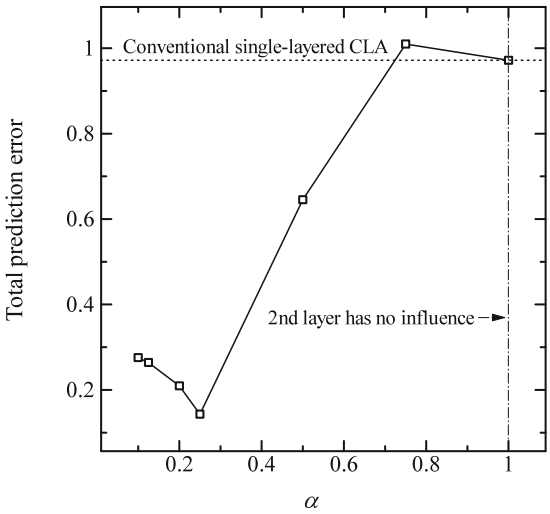

Fig. 3. Effects of parameter α to reflect the prediction from 2nd layer to the 1st layer on the combined sine $X_c(t, 2)$

5 Experimental Results and Discussion

5.1 Effects of Relaxing Parameter α

The proposed double-layered CLA has the relaxing parameter α to reflect the prediction in the second layer to the prediction in the first layer. The smaller α, the stronger influence from the second layer to the first layer.

Figure 3 shows the total prediction error of the proposed double-layered CLA on the combined sine $X_c(t, 2)$ in the last half of the entire time-steps when we vary the relaxing parameter α. $\alpha = 1$ indicates that the any prediction in the second layer does not reflect to the first layer. As a reference, the result of the

conventional single-layered CLA is plotted as a horizontal dot line in this figure. From the result, we see that the total prediction error decreases by decreasing α from 1. We see that there is the best parameter α^* minimizing the total prediction error. In this case, $\alpha^* = 0.25$. Further decrease of α from α^* increases the prediction errors since the influence from the second layer becomes too strong for the first layer. In the following experiments, we use $\alpha^* = 0.25$ for the proposed double-layered CLA.

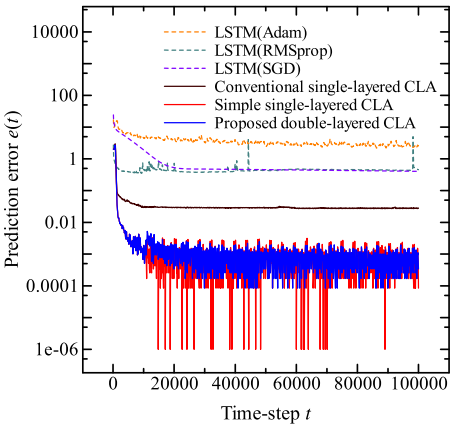

Fig. 4. Sine $X_s(t)$

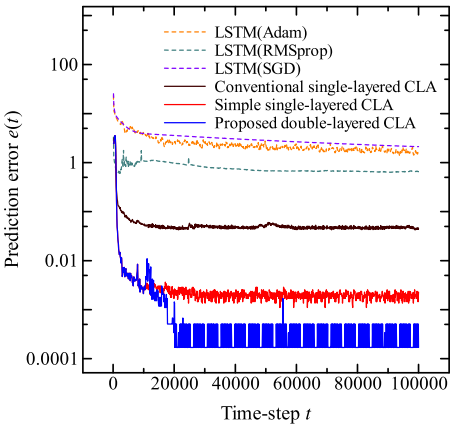

Fig. 5. Combined sine $X_c(t, 2)$

5.2 Prediction Accuracy

Figures 4, 5, 6, 7, 8, 9 show the histories of the prediction errors of the three LSTMs, the conventional single-layered CLA, the simple single-layered CLA and the proposed double-layered CLA on different benchmark time-series data.

Figure 4 shows the results on the most simple sin $X_s(t)$. In this case, we see that the prediction errors of the three CLAs are lower than those of the three LSTMs. In addition, we see that the simple CLA and the proposed CLA achieves lower prediction errors than the conventional CLA. On the other hand, we cannot see a clear difference between the simple single-layered CLA and the proposed double-layered CLA. However, we see that the cyclic increase of prediction error, especially in the case of the simple single-layered CLA. The proposed double-layered CLA suppresses the cyclic increase of the prediction errors on sin $X_s(t)$. This result reveals that the proposed double-layered CLA improves the prediction stability.

Figure 5 shows the results on the combined sine $X_c(t, 2)$. From the result, we see that the proposed double-layered CLA significantly decreases the prediction errors compared with the conventional single-layered CLAs and LSTMs. Figures 6, 7, 8 respectively show results on the combined sine $X_c(t, 3)$, $X_c(t, 4)$, and $X_c(t, 5)$. We see that the proposed double-layered CLA achieves lower prediction

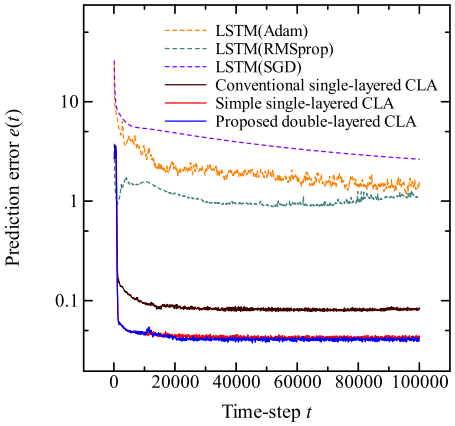

Fig. 6. Combined sine $X_c(t, 3)$

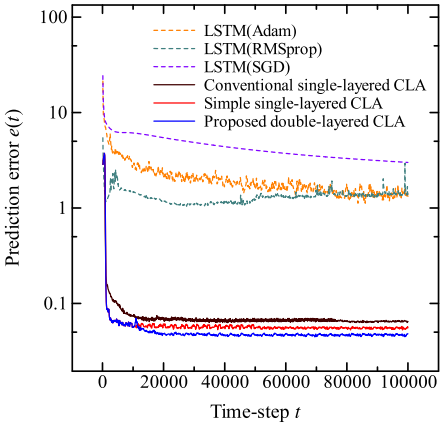

Fig. 7. Combined sine $X_c(t, 4)$

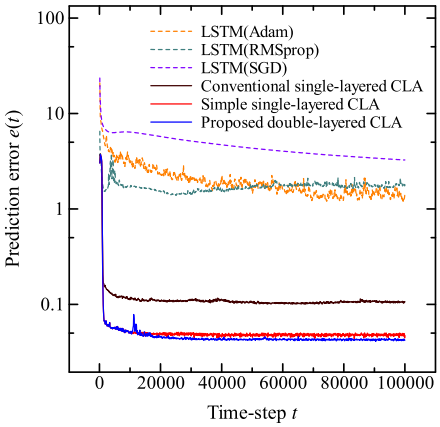

Fig. 8. Combined sine $X_c(t, 5)$

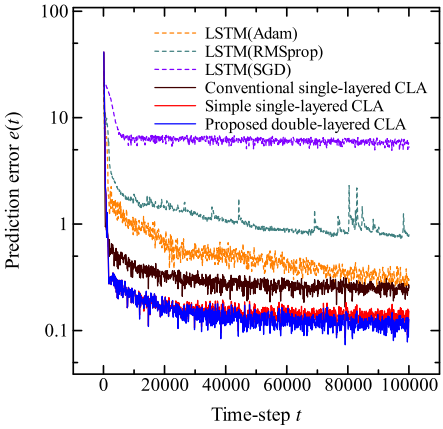

Fig. 9. Logistic mapping $X_l(t, 3.6)$

errors than the conventional single-layered CLAs and LSTMs. Figure 9 show the result on the time-series data on the logistic mapping. Also, in this case, we see a tendency that the proposed double-layered CLA achieves lower prediction errors than the conventional single-layered CLAs and LSTMs.

These results reveal that the double-layered CLA improves the prediction accuracy compared with the conventional single-layered CLA by accumulating the predictors of CLA and interacting between them based on the proposed interaction mechanism.

6 Conclusions

To improve the time-series prediction accuracy, in this work, we proposed a double-layered CLA. The proposed CLA receives input data and represents it and its context in the first layer. The input data context presentation in the first layer is then transferred to the second layer, and it is represented in the second layer as an abstract representation. Also, the abstract prediction in the second layer is reflected to the first layer to modify and enhance the prediction in the first layer. The experimental results using the benchmark time-series data show that the proposed double-layered CLA achieves lower prediction errors than the conventional single-layered CLAs and LSTMs.

As future works, we will accumulate more layers as multi-layered CLA and check the abstract data representation and prediction in the higher layers. As the drawback of proposed CLA, it has more parameters than conventional CLA. To solve this problem, we will consider the method of dynamic parameter setting.

Acknowledgments. This work was supported by JSPS KAKENHI Grant Number 20J14240.

References

1. Hawkins, J., Blakeslee, S.: On Intelligence. Times Books, New York (2004)
2. Ahmad, S., Hawkins, J.: Properties of Sparse Distributed Representations and their Application to Hierarchical Temporal Memory. Technical Report, pp. 1–18 (2015)
3. Zyarah, A.M., Kudithipudi, D.: Neuromemrisitive architecture of HTM with on-device learning and neurogenesis. ACM J. Emerg. Technol. Comput. Syst. **15**(3), 24 (2019). Article 24
4. Hawkins, J., Subutai, A., Dubinsky, D.: Hierarchical temporal memory including HTM cortical learning algorithms. Technical Report, Numenta Inc., (2010)
5. Hawkins, J., Subutai, A.: Why neurons have thousands of synapses, a theory of sequence memory in neocortex. Front. Neural Circuits **10**, 1–13 (2016)
6. https://github.com/numenta/nupic . Accessed 1 Mar 2021
7. Elman, J.L.: Finding structure in time. Cogn. Sci. **14**(2), 179–211 (1990)
8. Connor, J.T., Martin, R.D., Atlas, L.E.: Recurrent neural networks and robust time series prediction. IEEE Trans. Neural Networks **5**(2), 240–254 (1994)
9. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
10. Cui, Y., Ahmad, S., Hawkins, J.: Continuous online sequence learning with an unsupervised neural network model. Neural Comput. **28**(11), 2474–2504 (2016)
11. Aoki, T., Takadama, K., Sato, H.: A preliminary study on a multi-layered cortical learning algorithm. In: The 7th UEC Seminar in ASEAN, 2020 and The 2nd ASEAN-UEC Workshop on Energy and AI (2020)
12. Aoki, T., Takadama, K., Sato, H.: Study on simple cortical learning algorithm and prediction accuracy improvement. In: SSI 2017, The Society of Instrument and Control Engineers, pp. 135–140 (2017) (in Japanese)
13. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. CoRR arXiv:1412.6980 (2014)
14. Hornik, K.: Approximation capabilities of multilayer feedforward networks. Neural Network **4**(2), 251–257 (1991)
15. https://github.com/keras-team/keras. Accessed 30 June 2021