SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

September 2020, Vol. 63 190202:1–190202:11
https://doi.org/10.1007/s11432-019-2844-3

Special Focus on Driver Automation Collaboration and Augmentation in Automated Driving

Investigating the dynamic memory effect of human drivers via ON-LSTM

Shengzhe DAI[1,2], Zhiheng LI[1,2], Li LI[1*], Dongpu CAO[3], Xingyuan DAI[4] & Yilun LIN[4]

[1]*Department of Automation, Tsinghua University, Beijing* 100084*, China;*
[2]*Shenzhen International Graduate School, Tsinghua University, Shenzhen* 518055*, China;*
[3]*Department of Mechanical and Mechatronics Engineering, University of Waterloo, Waterloo* N2L 3G1*, Canada;*
[4]*State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing* 100080*, China*

Received 29 November 2019/Revised 7 February 2020/Accepted 29 February 2020/Published online 13 August 2020

Abstract It is a widely accepted view that considering the memory effects of historical information (driving operations) is beneficial for vehicle trajectory prediction models to improve prediction accuracy. However, many commonly used models (e.g., long short-term memory, LSTM) can only implicitly simulate memory effects, but lack effective mechanisms to capture memory effects from sequence data and estimate their effective time range (ETR). This shortage makes it hard to dynamically configure the most suitable length of used historical information according to the current driving behavior, which harms the good understanding of vehicle motion. To address this problem, we propose a modified trajectory prediction model based on ordered neuron LSTM (ON-LSTM). We demonstrate the feasibility of ETR estimation based on ON-LSTM and propose an ETR estimation method. We estimate the ETR of driving fluctuations and lane change operations on the NGSIM I-80 dataset. The experiment results prove that the proposed method can well capture the memory effects during trajectory prediction. Moreover, the estimated ETR values are in agreement with our intuitions.

Keywords driving behavior, memory effect, trajectory prediction, historical information, ON-LSTM

Citation Dai S Z, Li Z H, Li L, et al. Investigating the dynamic memory effect of human drivers via ON-LSTM. Sci China Inf Sci, 2020, 63(9): 190202, https://doi.org/10.1007/s11432-019-2844-3

1 Introduction

How to identify and understand vehicle driving behaviors has been a hot research issue for advanced driving assistance system (ADAS) and automated vehicles in the past few decades. Here, a driving behavior depicts a set of implicit but understandable rules that the driver reacts to the surrounding vehicles (environments) and performs targeted driving operations. The driving behaviors can be manually classified and named according to our understanding of the vehicle movements in the research scenario (e.g., going straight, lane change, overtaking, etc.). There are various specific tasks of ADAS and automated vehicles that heavily rely on driving behavior recognition, such as the vehicle trajectory prediction we focus on in this paper.

There is a commonly accepted view that considering historical information is crucial for behavior recognition and accurate trajectory prediction [1–4]. To support this view, many researchers have proven that an instantaneous driving operation can generate a long-term impact on vehicle behavior (movement), and this phenomenon is called the memory effect [1–3]. The memory effect is an implicit mechanism during

* Corresponding author (email: li-li@tsinghua.edu.cn)

© Science China Press and Springer-Verlag GmbH Germany, part of Springer Nature 2020

driving, which cannot be directly observed. It can only be indirectly reflected through some specially designed models or features.

Some early studies [5–7] did not consider historical information (nor memory effects) and only used the current observed states (e.g., positions, velocities, accelerations, etc.) for trajectory prediction. These models are easy to calculate, but they do not consider the uncertainties of the observed states [8] and cannot distinguish driving fluctuations[1)] from vehicle behavior changes due to the lack of historical information, resulting in unreliable long-term predictions.

In the last decade, researchers gradually realize the importance of historical information and memory effects. Almost all the recent studies utilize historical information, while some studies try various methods to simplify or simulate memory effects for more accurate predictions. Some studies adopt Gaussian process regression (GPR) to consider historical coordinates (but not to consider memory effects) of the target vehicle in trajectory calculations [9–11]. Some studies extract some statistics (e.g., RMS, mean, etc.) of the observed states in multiple time windows or design some heuristic features to simplify and quantize the memory effects [12–14]. Recently, LSTM [15–17] became a commonly used model for trajectory prediction because its gate structures allowed the long-term retention of information and thus could model long-term dependencies of temporal data [18–22]. This characteristic helps to implicitly simulate the memory effects during the trajectory prediction and further improves prediction accuracy.

However, an important question about memory effects remains unsolved: How to estimate the effective time range (ETR) of the memory effect generated by a specific driving operation[2)]? This question becomes crucial when we determine the time range of the input historical information for behavior recognition and trajectory prediction[3)]. We hope that we can automatically obtain the most appropriate range that best suits the studied behavior by estimating the ETR of the corresponding memory effect. However, there has not been a mature ETR estimation method yet; thus most of the recent studies can only find a relatively suitable range by enumerating and testing various candidate values [21,23]. This selection process is time consuming and the selected range cannot be directly generalized to other datasets. Moreover, this value is usually overlarge to ensure that sufficient historical information can be provided for most behaviors, but actually, the provided historical information is redundant for many behaviors.

As described above, the commonly used LSTM can implicitly simulate the memory effects through its special gate structures. However, two existing problems make it difficult for LSTM to estimate ETR of memory effects. First, LSTM cannot capture the start time of a certain memory effect by analyzing some characteristics (e.g., some special change laws) of its gate values. Second, LSTM cannot evaluate the correlation degree of information between two sequence points to determine the end time of a memory effect.

There is a common reason for the above two problems. Since the calculation process of LSTM is a "black box", both the internal variables and gate values of LSTM cells lack semantics and thus cannot be used to analyze memory effects [24]. To solve the above problems, inspired by the good application of ordered neuron LSTM (ON-LSTM) on unsupervised constituency parsing [24], we modify our previous trajectory prediction model for a single vehicle (i.e., the C_s in [25]) by adopting ON-LSTM instead of classical LSTM and propose an ETR estimation method in this paper.

The ON-LSTM model introduces the concept of information levels (i.e., a kind of semantic) to distinguish the information that influences different time ranges. The corresponding hierarchical updating logic of ON-LSTM cells maintains the semantics of levels during the calculation process, which enables us to analyze the correlation degrees between adjacent input information (in Section 2). Moreover, the relationship between memory effects and the change laws of the gate values of ON-LSTM cells will be discussed in Subsection 3.2. These two characteristics of ON-LSTM well support the feasibility of ETR

1) In this paper, a driving fluctuation refers to a slight driving operation that does not aim to change the vehicle behavior (e.g., the lateral rectification when the vehicle approaches an adjacent lane during going straight).

2) In this paper, the term driving operation includes both targeted behavior change operations and driving fluctuations.

3) An appropriate range of historical information is important for behavior recognition and trajectory prediction. If the range is too small, the historical information is not sufficient to express the characteristics of the current behavior. If the range is too large, the earlier information is not effective (even harmful) to assist the current recognition and prediction due to the weak correlation.

estimation.

We train the modified model and evaluate our ETR estimation method on the NGSIM I-80 dataset. Specifically, we respectively evaluate the ETR of the memory effects caused by the fluctuations during going straight movements and lane change operations. The experiment demonstrates that the variations of the gate values in ON-LSTM models can well capture the driving operations (i.e., the start of the memory effect). Moreover, the estimated ETR values of the different driving behaviors have significant differences but are in agreement with our intuitions.

The main contributions of this paper can be summarized as follows.

• We propose a novel ETR estimation method based on ON-LSTM for vehicle memory effects during driving. This method helps to better understand the mechanism of memory effects and to set more appropriate horizons of historical information for the trajectory prediction problem.

• The proposed method does not increase the time complexity nor the space complexity of the model and can run both online and offline. Therefore, this method can be easily integrated into LSTM-based models.

To better present our findings, the rest of this paper is arranged as follows. Section 2 describes the structure, calculations, and applications of the ON-LSTM model. Section 3 describes the method to estimate ETR and presents the results of the experiment. Finally, Section 4 concludes the paper.

2 The structure and comprehension of ON-LSTM

The ON-LSTM model is derived from the classical LSTM model by modifying the basic structure of LSTM cells [24]. It not only inherits the ability of classical LSTM to memorize historical information through the gate structures and further to simulate the memory effect of sequences, but also introduces the concept of "information levels" in variables to construct a hierarchical updating logic for information and simulate more complex memory effects. Moreover, its structure enables us to analyze the correlation degrees between sequence points by observing the gate values. This new function is important for ETR estimation.

In this section, we will introduce the detailed structure and advantages of ON-LSTM. In Subsection 2.1, we will present the structure and the calculations of basic ON-LSTM cells. In Subsection 2.2, we will first propose the method to evaluate correlation degrees. Then we will describe the typical usages of ON-LSTM and introduce its application in this paper.

2.1 The structure of ON-LSTM cells

In this subsection, we will compare the different structures and calculations between classical LSTM cells and ON-LSTM cells.

The classical LSTM cell. The classical LSTM cell is a basic memory unit of LSTM. It also allows LSTM to automatically forget some historical information, incorporate the new information, and generate output vector when new sequence data vector $\boldsymbol{x}_t$ is input at the time t. The above three functions are respectively controlled by three gate vectors: the forget gate $\boldsymbol{f}_t$, the input gate $\boldsymbol{i}_t$ and the output gate $\boldsymbol{o}_t$. The calculation of the updating process is formulated by

$$\begin{cases} \boldsymbol{f}_t = \sigma\left(\boldsymbol{W}_f \boldsymbol{x}_t + \boldsymbol{U}_f \boldsymbol{h}_{t-1} + \boldsymbol{b}_f\right), \\ \boldsymbol{i}_t = \sigma\left(\boldsymbol{W}_i \boldsymbol{x}_t + \boldsymbol{U}_i \boldsymbol{h}_{t-1} + \boldsymbol{b}_t\right), \\ \boldsymbol{o}_t = \sigma\left(\boldsymbol{W}_o \boldsymbol{x}_t + \boldsymbol{U}_o \boldsymbol{h}_{t-1} + \boldsymbol{b}_i\right), \\ \hat{\boldsymbol{c}}_t = \tanh\left(\boldsymbol{W}_c \boldsymbol{x}_t + \boldsymbol{U}_c \boldsymbol{h}_{t-1} + \boldsymbol{b}_c\right), \\ \boldsymbol{c}_t = \boldsymbol{f}_t \circ \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \circ \hat{\boldsymbol{c}}_t, \\ \boldsymbol{h}_t = \boldsymbol{o}_t \circ \tanh\left(\boldsymbol{c}_t\right), \end{cases} \tag{1}$$

where σ is the Sigmoid function. The operator "$\circ$" means element-wise product. The parameters $\boldsymbol{W}_f$, $\boldsymbol{U}_f$ and $\boldsymbol{b}_f$ ($\boldsymbol{W}_i$, $\boldsymbol{W}_o$, $\boldsymbol{W}_c$, $\boldsymbol{U}_i$, $\boldsymbol{U}_o$, $\boldsymbol{U}_c$, $\boldsymbol{b}_i$, $\boldsymbol{b}_o$, $\boldsymbol{b}_c$) are the learned weight matrices and bias vectors for

$\boldsymbol{f}_t$ ($\boldsymbol{i}_t$, $\boldsymbol{o}_t$, $\hat{\boldsymbol{c}}_t$). The calculation of $\boldsymbol{c}_t$ expresses the fusion between the valid historical information $\boldsymbol{c}_{t-1}$ (filtered by $\boldsymbol{f}_t$) and the new input information (filtered by $\boldsymbol{i}_t$). It also serves as the memorized historical information for the next time step. The vector $\boldsymbol{h}_t$ defines the output vector controlled by $\boldsymbol{o}_t$.

The ON-LSTM cell. The elements of the vectors in classical LSTM cells are unordered and lack of semantics. Here, we directly introduce the term "ordered" from [24] and retain its original meaning in [24]. In this paper, an ordered vector means that the vector elements have some semantic continuity based on their locations in the vector, rather than that the elements have been sorted according to some measures (such as the numerical values).

In the classical LSTM, since the semantic of vector order is not explicitly defined, the learned knowledge of vector elements is relatively random (and different elements may learn similar things), which actually waste some potentials of the occupied space. In ON-LSTM cells, the semantic of order is defined as information level, which distinguishes and stores the information that influences different ranges of time: high-level elements store long-term information which is kept for a long time, while low-level elements store short-term information that can be rapidly forgotten [24]. This definition supports ON-LSTM to learn extra knowledge of ordered semantics without increasing the space complexity of the LSTM network. Moreover, this hierarchical structure allows us to separately process the transmission, integration, and forgetting of the information at different levels and further to simulate a more complex memory effect.

Following the above idea, all the gate vectors (i.e., $\boldsymbol{f}_t$, $\boldsymbol{i}_t$, $\boldsymbol{o}_t$) and (hidden) state vectors (i.e., $\boldsymbol{h}_t$, $\hat{\boldsymbol{c}}_t$, $\boldsymbol{c}_t$) become ordered vectors. Without loss of generality, we set the latter elements in the vectors as the memory unit for higher-level information.

To organize the calculation of the hierarchical updating process, ON-LSTM designs another two gates: the master forget gate $\tilde{\boldsymbol{f}}_t$ and the master input gate $\tilde{\boldsymbol{i}}_t$. These two gate vectors are actually masks to select whether a specific level should follow the historical information or adopt the new input information or fuse them together by certain weights. The two gate vectors are calculated by

$$\begin{cases} \tilde{\boldsymbol{f}}_t = \overrightarrow{\mathrm{cs}}\left(\mathrm{softmax}\left(\boldsymbol{W}_{\tilde{f}}\boldsymbol{x}_t + \boldsymbol{U}_{\tilde{f}}\boldsymbol{h}_{t-1} + \boldsymbol{b}_{\tilde{f}}\right)\right), \\ \tilde{\boldsymbol{i}}_t = \overleftarrow{\mathrm{cs}}\left(\mathrm{softmax}\left(\boldsymbol{W}_{\tilde{i}}\boldsymbol{x}_t + \boldsymbol{U}_{\tilde{i}}\boldsymbol{h}_{t-1} + \boldsymbol{b}_{\tilde{i}}\right)\right), \end{cases} \tag{2}$$

where $\boldsymbol{W}_{\tilde{f}}$, $\boldsymbol{W}_{\tilde{i}}$, $\boldsymbol{U}_{\tilde{f}}$, $\boldsymbol{U}_{\tilde{i}}$, $\boldsymbol{b}_{\tilde{f}}$ and $\boldsymbol{b}_{\tilde{i}}$ are the learned weight matrixes and bias vectors. $\overrightarrow{\mathrm{cs}}(\boldsymbol{v})$ and $\overleftarrow{\mathrm{cs}}(\boldsymbol{v})$ respectively define a rightward and a leftward cumulative sum function. Suppose $\boldsymbol{v} = [v_1, v_2, \ldots, v_n]$, and then $\overrightarrow{\mathrm{cs}}(\boldsymbol{v}) = [v_1, (v_1 + v_2), \ldots, \sum_{i=1}^{n} v_i]$ and $\overleftarrow{\mathrm{cs}}(\boldsymbol{v}) = [\sum_{i=1}^{n} v_i, \sum_{i=1}^{n-1} v_i, \ldots, v_1]$.

It is straightforward that the elements on the right (left) side of $\tilde{\boldsymbol{f}}_t$ ($\tilde{\boldsymbol{i}}_t$) approach to 1, so the higher-level (lower-level) information can be activated by $\tilde{\boldsymbol{f}}_t$ ($\tilde{\boldsymbol{i}}_t$). Moreover, another generated gate vector $\boldsymbol{\omega}_t = \tilde{\boldsymbol{f}}_t \circ \tilde{\boldsymbol{i}}_t$ is used to activate the middle-level information. Therefore, the calculation of $\boldsymbol{c}_t$ is formulated by

$$\boldsymbol{c}_t = \boldsymbol{\omega}_t \circ (\boldsymbol{f}_t \circ \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \circ \hat{\boldsymbol{c}}_t) + \left(\tilde{\boldsymbol{f}}_t - \boldsymbol{\omega}_t\right) \circ \boldsymbol{c}_{t-1} + \left(\tilde{\boldsymbol{i}}_t - \boldsymbol{\omega}_t\right) \circ \hat{\boldsymbol{c}}_t. \tag{3}$$

Eq. (3) and Figure 1(a) express three updating logics of $\boldsymbol{c}_t$. The first item updates the middle-level information by fusing the historical information and the new input information, just like the classical LSTM cell. The second item updates the high-level information by directly adopting the historical information since it is sufficient to depict long-term memory effects. In contrast, the third item updates the low-level information only by the new input information.

Suppose $\tilde{\boldsymbol{f}}_t = [\tilde{f}_1, \ldots, \tilde{f}_n]$ and $\tilde{\boldsymbol{i}}_t = [\tilde{i}_1, \ldots, \tilde{i}_n]$. f_{mas} is the lowest level activated by $\tilde{\boldsymbol{f}}_t$, and i_{mas} is the highest level activated by $\tilde{\boldsymbol{i}}_t$. We can find that $\boldsymbol{\omega}_t$ approximates to a zero vector when $f_{\mathrm{mas}} > i_{\mathrm{mas}}$. Therefore, the elements in the middle levels of $\boldsymbol{c}_t$ are not activated, as shown in Figure 1(b).

To sum up, the complete hierarchical updating logic of ordered information is formulated by (4). It is straightforward that the introduce of "order" does not increase the computational complexity of cell

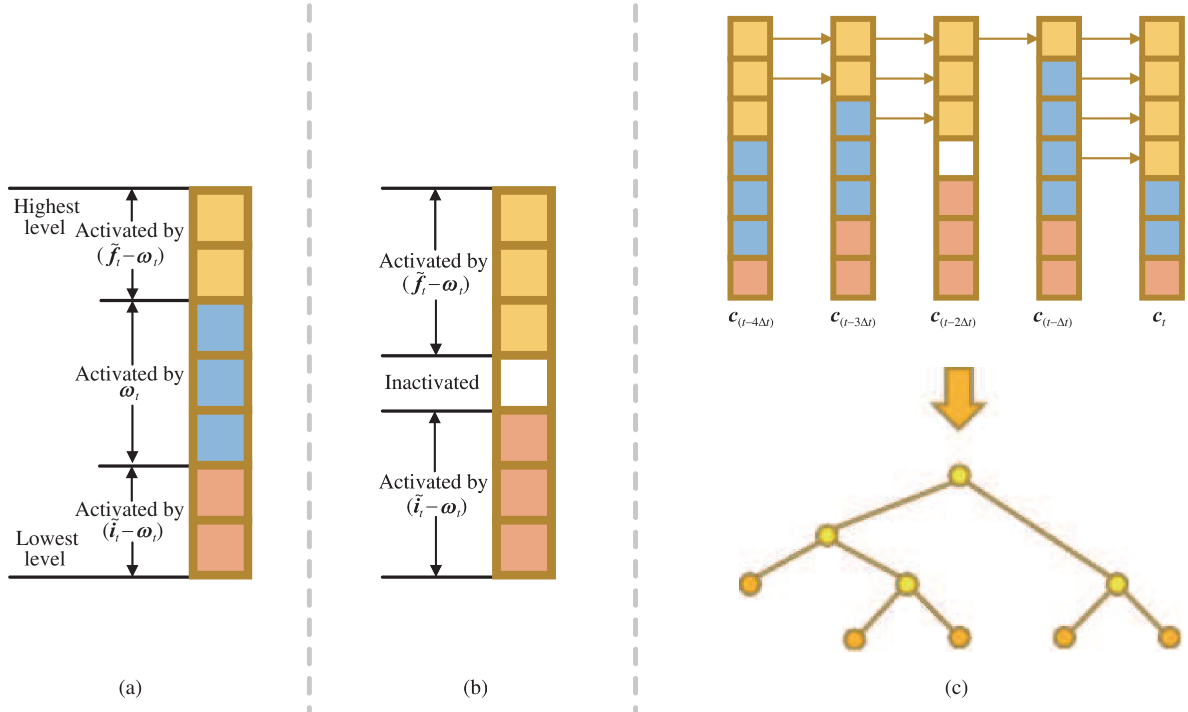

Figure 1 (Color online) The illustration of the hierarchical updating logic of ON-LSTM. (a) The hierarchical update when $\boldsymbol{\omega}_t$ does not approximate to zero vector; (b) the hierarchical update when $\boldsymbol{\omega}_t$ approximates to zero vector; (c) the hierarchical updating process and the corresponding constituency tree of a specific sequence.

updating.

$$\begin{cases} \boldsymbol{f}_t = \sigma\left(\boldsymbol{W}_f\boldsymbol{x}_t + \boldsymbol{U}_f\boldsymbol{h}_{t-1} + \boldsymbol{b}_f\right), \\ \boldsymbol{i}_t = \sigma\left(\boldsymbol{W}_i\boldsymbol{x}_t + \boldsymbol{U}_i\boldsymbol{h}_{t-1} + \boldsymbol{b}_t\right), \\ \boldsymbol{o}_t = \sigma\left(\boldsymbol{W}_o\boldsymbol{x}_t + \boldsymbol{U}_o\boldsymbol{h}_{t-1} + \boldsymbol{b}_i\right), \\ \hat{\boldsymbol{c}}_t = \tanh\left(\boldsymbol{W}_c\boldsymbol{x}_t + \boldsymbol{U}_c\boldsymbol{h}_{t-1} + \boldsymbol{b}_c\right), \\ \tilde{\boldsymbol{f}}_t = \overrightarrow{\mathrm{cs}}\left(\mathrm{softmax}\left(\boldsymbol{W}_{\tilde{f}}\boldsymbol{x}_t + \boldsymbol{U}_{\tilde{f}}\boldsymbol{h}_{t-1} + \boldsymbol{b}_{\tilde{f}}\right)\right), \\ \tilde{\boldsymbol{i}}_t = \overleftarrow{\mathrm{cs}}\left(\mathrm{softmax}\left(\boldsymbol{W}_{\tilde{i}}\boldsymbol{x}_t + \boldsymbol{U}_{\tilde{i}}\boldsymbol{h}_{t-1} + \boldsymbol{b}_{\tilde{i}}\right)\right), \\ \boldsymbol{c}_t = \boldsymbol{\omega}_t \circ \left(\boldsymbol{f}_t \circ \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \circ \hat{\boldsymbol{c}}_t\right) + \left(\tilde{\boldsymbol{f}}_t - \boldsymbol{\omega}_t\right) \circ \boldsymbol{c}_{t-1} + \left(\tilde{\boldsymbol{i}}_t - \boldsymbol{\omega}_t\right) \circ \hat{\boldsymbol{c}}_t, \\ \boldsymbol{h}_t = \boldsymbol{o}_t \circ \tanh\left(\boldsymbol{c}_t\right). \end{cases} \tag{4}$$

2.2 The advantages and applications of ON-LSTM models

According to Subsection 2.1, the main modification of ON-LSTM lies in that we attach a constraint with specific semantics to the memory effects of the model. This modification allows the model to explicitly distinguish the information that influences different ranges of time, which cannot be learned in classical LSTM. Moreover, this modification provides the capability to track the introduction time of a certain level information by observing the values of historical $\tilde{\boldsymbol{f}}_t$ and $\tilde{\boldsymbol{i}}_t$ (f_{mas} and i_{mas}).

We simulate the hierarchical updating process of a sequence in Figure 1(c). According to the figure, we can clearly find that the second to fourth highest-level information of $\boldsymbol{c}_t$ is mainly brought from the input at the time $(t-\Delta t)$. Thus the elements at the time $(t-\Delta t)$ and t in this sequence are demonstrated to have a strong correlation while lacking correlation with earlier data. So we can split the original sequence at $(t-\Delta t)$ to obtain two relatively independent subsequences. It is straightforward that a suitable segmentation point is usually at the local peak of f_{mas} since most low-level elements tend to forget historical information.

Similarly, we can continue to segment the sequence based on the above method until obtaining the smallest units (i.e., single elements). During performing segmentation, we can build a tree-like graph to record the relevance between different subsequences, as shown in Figure 1(c). This tree is proved to be a reasonable constituency parsing, which helps to understand the semantics or grammars of sequence data such as sentences [24].

However, the sequence data (i.e., vehicle trajectories) studied in this paper do not have clear grammatical bounds to split different components (driving motions), so we cannot build a parse tree for vehicle trajectories like [24]. But we can still use the above method to evaluate the correlation degree between two adjacent trajectory points, which is a core method for the ETR estimation task.

3 The ETR estimation method and experiments

In this section, we will demonstrate that the ON-LSTM model can dynamically capture the memory effect when a driving operation is performed. Specifically, the start time of memory effect is observable through the variations of f_{mas} and i_{mas}, which is another core method for ETR estimation. Our ETR estimation method is established based on the above two basic methods.

In this section, we first present the training of the ON-LSTM trajectory prediction model (in Subsection 3.1) to support the following ETR estimation. Then we propose the method to capture memory effects by observing the change laws of gate values of the learned ON-LSTM in Subsection 3.2. Finally, we propose the detailed ETR estimation method and present the testing results in Subsection 3.3.

3.1 The model, dataset, and experiment settings

The trajectory prediction model. The model we use in this paper is modified from our previous trajectory prediction model (i.e., the single-vehicle model C_s in [25])[4]. We adopt ON-LSTM instead of classical LSTM to introduce the hierarchical updating logic for multiple information levels, which helps to simulate a more complex process of memory effects. Thus ideally, the modified model will yield more accurate trajectory predictions than the original model. Moreover, the modified model will support the following ETR estimation method.

Note that we only modify the LSTM cells, while keeping the original model structure and hyper-parameters of C_s. Because the computational complexities of LSTM cells and ON-LSTM cells are the same, the new trajectory prediction model has a similar prediction consuming time to C_s.

The input and output of the model. The input of the trajectory prediction model is a set of sequences that contains the lateral and longitudinal coordinates of the target vehicle within a historical time horizon t_h. The output is a predicted trajectory sequence within a prediction time horizon t_p. During training, these time horizons are set as $t_h = 3$ s and $t_p = 6$ s, which follow the values in [25].

The dataset and preprocessing. We train and evaluate our model on the NGSIM I-80 dataset. The research scenario is a straight road with multiple lanes. As we discussed in [25], the NGSIM I-80 is a severely unbalanced dataset, so we perform dataset balancing trajectory trimming as we did in [25].

To evaluate the function of ETR estimation, we build a testing set that contains 3000 trajectory segments with a length of 8 s (80 points with the time resolution of 0.1 s). We manually label the driving behaviors of all the trajectory points in the obtained testing set. The behavior categories include five types: going straight (GS), left lane change (LLC), merging into the left lane (MLL), right lane change (RLC), and merging into the right lane (MRL). Here, the transition region between GS and LLC/RLC is considered to have a real lane change operation.

Model training and testing. The training process and hyper-parameter values of our model are the same as those of C_s in [25]. During testing, the first 3 seconds' data in each trajectory segment serve as the initial historical information. Then we gradually perform the calculation at each time step and record the values of $\tilde{\boldsymbol{f}}_t$ ($\tilde{\boldsymbol{i}}_t$) in each ON-LSTM cell. Finally, we analyze the variation process of $\tilde{\boldsymbol{f}}_t$ ($\tilde{\boldsymbol{i}}_t$) to

4) In this section, we only introduce the differences between the two models and some major details. The omitted specific model structure, the preprocessing of the dataset, the method to select hyper-parameters, the training process, and the overfitting analysis method are presented in [25].

Table 1 RMS values of prediction error

Prediction horizon (s)	RMS value (m)	
	C-LSTM	ON-LSTM
1	0.54	0.48
2	1.17	1.04
3	1.90	1.71
4	2.73	2.50
5	3.67	3.43
6	4.70	4.40

capture driving operations and estimate the corresponding ETR, whose detailed method will be proposed in Subsection 3.2.

The RMS prediction error of the trained model. We compare the RMS prediction error between our model (denoted as ON-LSTM) with our previous single-vehicle model based on classical LSTM models (denoted as C-LSTM). The definition and calculation of the RMS error are the same as those in [25]. The RMS testing result is shown in Table 1 and indicates that the ON-LSTM yields more accurate predictions than C-LSTM. This result is in line with our intuition that the more complex memory effects simulated by ON-LSTM benefit to the prediction task.

We also calculate the prediction consuming time of the two models. Specifically, the average time of C-LSTM and ON-LSTM to perform a single trajectory prediction with the horizon of 6 s is respectively 1.07 ms (C-LSTM) and 1.19 ms (ON-LSTM) on a TITAN X (Pascal) GPU. Therefore, the model based on ON-LSTM is sufficient to execute a real-time trajectory prediction.

3.2 Capturing memory effects based on ON-LSTM

We have demonstrated that ON-LSTM can estimate the end time of a memory effect by analyzing the correlation degrees between adjacent points (in Subsection 2.2). In this subsection, we will prove that ON-LSTM can also estimate the start time of an impact (i.e., the corresponding driving operation) by observing some special variation laws of $f_{\rm mas}$ and $i_{\rm mas}$[5]. That is, the feasibility of ETR estimation is theoretically proven.

We find that the values of $f_{\rm mas}$ and $i_{\rm mas}$ have two change laws during the calculation of the trajectory prediction model. When the driving of the target vehicle is relatively stable (and the trajectory is smooth), the variations of $f_{\rm mas}$ and $i_{\rm mas}$ are usually slight.

When a certain driving operation happens, the varations of $f_{\rm mas}$ and $i_{\rm mas}$ present a "forgetting-rebuilding" process. Specifically, the value of $f_{\rm mas}$ first reaches a local peak to forget more long-term historical information (i.e., knowledge of the previous vehicle motion). Then $f_{\rm mas}$ drops rapidly and keeps $f_{\rm mas} < i_{\rm mas}$ for some time[6], so ω_t can continuously activate some middle information layers. During this time, the ON-LSTM model quickly rebuilds a new understanding of the current characteristic of vehicle motion by fusing both historical information and new information (i.e., simulating the memory effect). After the model accumulates a certain amount of knowledge of the current motion, $f_{\rm mas}$ will gradually rise and $i_{\rm mas}$ will gradually drop. In this paper, we can capture such "forgetting-rebuilding" processes to locate the start time of memory effects.

We randomly select a trajectory segment that contains an active lane change operation, record the values of $f_{\rm mas}$ and $i_{\rm mas}$ during the calculation, and plot the variation processes in Figure 2. In the selected trajectory, the manually labeled boundary between GS and LLC is at 2.9 s[7]. We can capture five local peaks (A_1–A_5) that may indicate the start of a memory effect in Figure 2. The target vehicle keeps a

5) Note that $f_{\rm mas}$ and $i_{\rm mas}$ in this paper are from the ON-LSTM cells in the first layer of the overall trajectory prediction model. However, we have tested ON-LSTM cells in the latter layers and find similar variation laws of those variables.

6) In most of the captured "forgetting-rebuilding" processes, the decrease of $f_{\rm mas}$ is usually accompanied by an obvious increase of $i_{\rm mas}$, such as the variations after A_1 and A_3 in Figure 2. However, the increase of $i_{\rm mas}$ is not a requirement, so the variations after A_4 in Figure 2 still indicate a valid "forgetting-rebuilding" process.

7) This value just follows our prior knowledge and has an inestimable deviation compared with the actual lane change operation time.

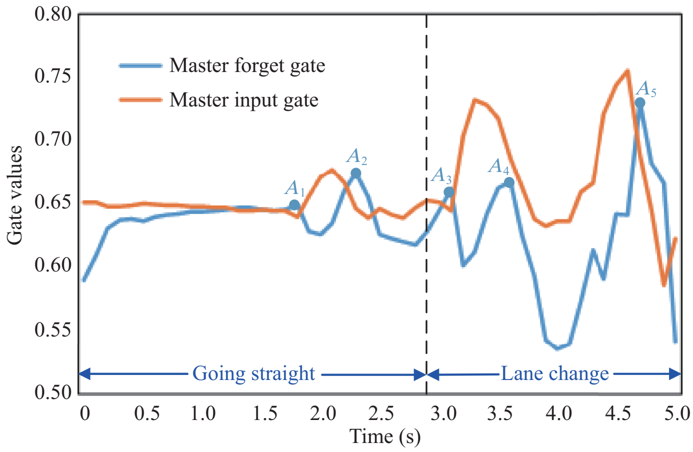

Figure 2 (Color online) The variations of $f_{\rm mas}$ and $i_{\rm mas}$ before and after a lane change operation is performed. Note that the values in the figure have been normalized.

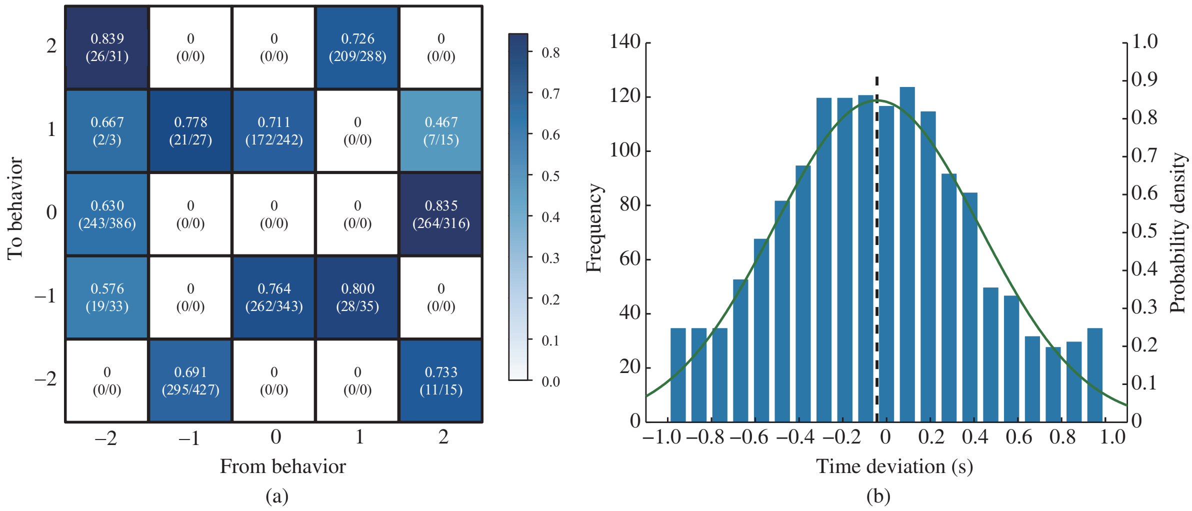

Figure 3 (Color online) The statistics of the captured memory effects in the testing set. (a) The proportion of significant memory effects of different behavior change relationships. The coordinate values in the figure respectively indicate different driving behaviors, specifically, MLL: −2; LLC: −1; GS: 0; RLC: 1; MRL: 2. (b) The distribution of the time deviation between the labeled behavior change point and the estimated start time of the memory effect.

relatively stable GS behavior at 0–1.7 s. During the period of 1.7–2.9 s, the vehicle's movement starts to differ from the previous GS and the values of $f_{\rm mas}$ and $i_{\rm mas}$ also indicate the occurrence of LLC (i.e., the fluctuations after A_1 and A_2). During the period of 2.9–5 s, the target vehicle continues the LLC behavior and the fluctuation amplitudes are much larger than those in 1.7–2.9 s, showing more intense "forgetting-rebuilding" processes.

In the above example, we can capture a typical memory effect (i.e., the fluctuation after A_3) near the labeled boundary between different driving behaviors[8)]. In fact, this is a common phenomenon for vehicle trajectories with driving behavior changes. In this paper, we label a total of 2161 behavior change moments from the 3000 trajectories in the testing set. We successfully capture significant memory effects within 1 s before or after 72.1% behavior change moments, as shown in Figure 3(a). Here, a significant memory effect requires that the value of $f_{\rm mas}$ declines (or $i_{\rm mas}$ increases) more than 10% and keeps $f_{\rm mas} < i_{\rm mas}$ for a while.

Figure 3(b) shows that the distribution of the time deviation between the labeled behavior change time T_l and the estimated start time T_e of a nearby memory effect approximates to a Gaussian distribution.

8) For the present, we suppose that this manually labeled boundary approximates to the real start time of the corresponding behavior change operation.

The standard deviation of this distribution is not a small value. Moreover, in Figure 2, we can find that the ON-LSTM model has already started to learn the knowledge of LLC since more than 1 s before T_l (i.e., the time of A_1), which indicates that the driver may have already performed some operations since that time. These two facts indirectly prove that manual labeling is not an accurate way for finding the real start time of driving operations (denoted by T_g). Due to the deviation between T_l and T_g, it is also not guaranteed to capture a valid memory effect within 1 s before and after T_l, even if valid memory effects do exist nearby. Therefore, the percentages in Figure 3(a) cannot be simply understood as recognition accuracies[9].

Furthermore, we cannot study the duration (of the motion changes) nor the corresponding ETR (of the operations) by a supervised method. Fortunately, the above ON-LSTM-based approach does not require supervision, and the successful capture of A_1 and A_2 proves the feasibility of this unsupervised approach to estimate the ETR generated by driving operations.

ON-LSTM models can also capture driving fluctuations. In our testing set, most of the captured memory effects are caused by the lateral rectifications during GS, while the corresponding operations are usually too slight to be manually labeled. We perform similar experiments and prove that ON-LSTM is also capable to deal with these slight operations.

3.3 The ETR estimation based on ON-LSTM

We have already presented the two necessary approaches for ETR estimation. Specifically, we introduce the method to capture a certain memory effect and estimate its start time by observing the variations of f_{mas} and i_{mas} (in Subsection 3.2). We also introduce the method to evaluate the correlation degree between two adjacent trajectory (sequence) points based on ON-LSTM, which is used to estimate the end time of the memory effect caused by a driving operation (in Subsection 2.2). In this subsection, we combine these two approaches for ETR estimation.

In the testing set, we capture a total of 7475 significant memory effects that are caused by driving fluctuations from the 3000 trajectories. 7171 of them are in the GS behavior. Figure 4 shows the statistics of the estimated ETR of these captured driving fluctuations. Here, the estimated ETR values are grouped according to the peak value of f_{mas} (denoted by F_{mas}) at the beginning of the memory effect.

As shown in Figure 4, the mean ETR of the driving fluctuations is 0.9 s. Moreover, we find that with the increasing of F_{mas}, the subsequent descent rate of f_{mas} usually gets larger (as shown in Figure 4(f)) and the estimated ETR of the corresponding driving operation gets smaller (as shown in Figure 4(a)–(e)). To explain this change, we traverse the testing set to observe the memory effects that begin with large F_{mas} values. We find most of these trajectories have frequent jitters, which indicates that frequent driving operations (active adjustments) are performed to maintain the current behavior (such as GS). In those cases, subsequent adjustments will continuously interrupt the previous memory effect and start a new one, thus shortening the ETR of each memory effect.

On the other hand, frequent adjustments are more easy to be recognized by the ON-LSTM model than mild driving fluctuations. Thus the model will automatically adopt a more aggressive measure (i.e., the above phenomenon) to rapidly learn the knowledge of the new motion, which also reduces the value of ETR.

Similarly, we estimate the ETR values of the captured lane change operations in Subsection 3.2 (as shown in Figure 5). The mean ETR of these operations is 1.4 s and is larger than that of driving fluctuations. This result is in agreement with our intuition, because the behavior changes during driving are a longer-term process than slight fluctuations, so the corresponding ETR value should also be larger.

According to Figures 4 and 5, the ETR values of most captured memory effects are less than 3 s. Thus when studying the trajectory prediction for straight road scenarios (i.e., only considering the five driving behaviors defined in this paper), a 3 s (or even less) range of historical information is sufficient to understand most vehicle motions and perform accurate trajectory prediction. However, we cannot ignore

9) It just intuitively illustrates a kind of consistency between the behavior changes we understand (based on prior knowledge) and the capturing results of memory effects.

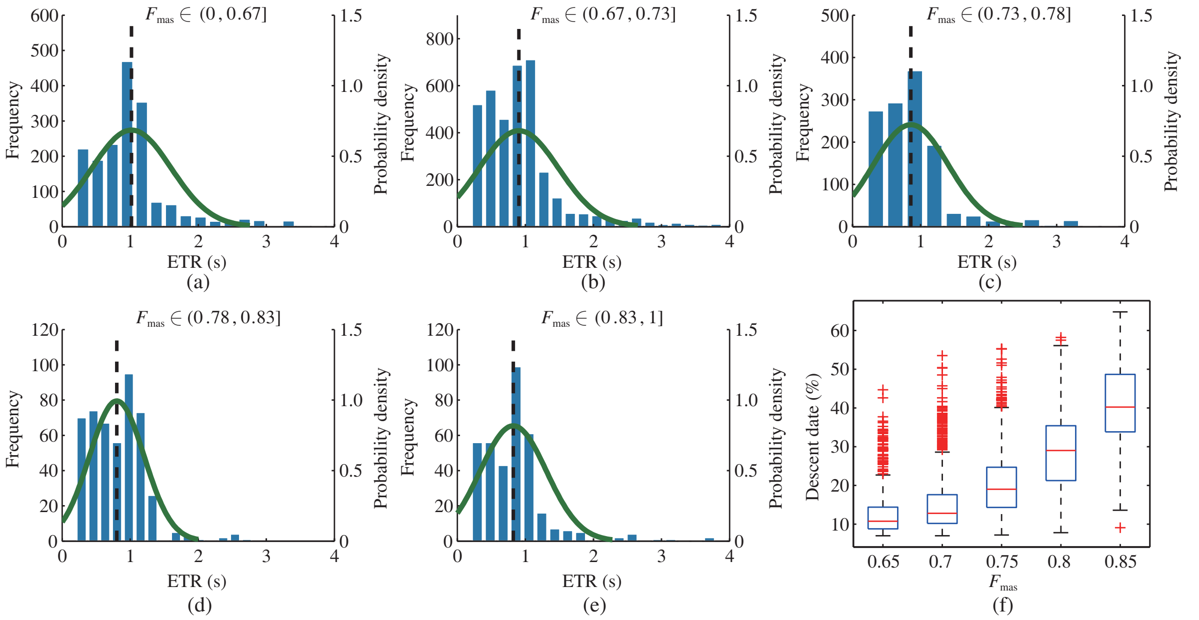

Figure 4 (Color online) The statistics of the estimated ETR of the captured driving fluctuations (grouped by the F_{mas} at the beginning of the memory effect). (a) $F_{mas} \in (0, 0.67]$, 1760 cases in total; (b) $F_{mas} \in (0.67, 0.73]$, 3639 cases in total; (c) $F_{mas} \in (0.73, 0.78]$, 1240 cases in total; (d) $F_{mas} \in (0.78, 0.83]$, 477 cases in total; (e) $F_{mas} \in (0.83, 1]$, 359 cases in total; (f) the boxplot of the descent rate of f_{mas} during the memory effect.

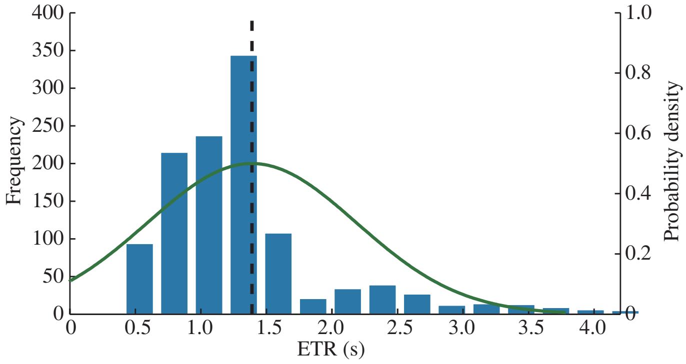

Figure 5 (Color online) The statistics of the estimated ETR of the captured motion change operations.

that the ETR values in some cases are still very large. Some current studies attempt to set a large range of historical information to cover these cases [23], while the overlarge range may harm the prediction of some cases with very small ETR. In contrast, our ETR estimation method could flexibly explore the best ranges for different behaviors, which helps to avoid the above shortcomings.

Moreover, the above statistics are performed offline (i.e., ETR is estimated after a complete trajectory is provided), but our method can also be run online (i.e., ETR of the current memory effect is estimated after each trajectory point is input). Therefore, our ETR estimation method also enables us to dynamically adjust the range of historical information according to the current situation.

4 Conclusion

In this paper, we study the memory effects generated by driving operations during vehicle movements. We focus on the difficulty of estimating the ETR and propose an estimation method based on the ON-LSTM model. The experiment on the NGSIM I-80 dataset proves that the estimated ETR values are in agreement with our intuitions. Moreover, the trajectory prediction model based on ON-LSTM yields more accurate predictions than the classical LSTM model.

However, we only test our method on NGSIM I-80, which only contains going straight and lane change movements. In the future work, we will perform ETR estimations on more types of vehicle movements to verify the effectiveness of this method.

Acknowledgements This work was supported in part by National Key Research and Development Program of China (Grant No. 2018AAA0101400), National Natural Science Foundation of China (Grant No. 61790565), Science and Technology Innovation Committee of Shenzhen (Grant No. JCYJ20170818092931604), and Intel Collaborative Research Institute for Intelligent and Automated Connected Vehicles (Grant No. ICRI-IACV).

References

1 Wang X, Jiang R, Li L, et al. Long memory is important: a test study on deep-learning based car-following model. Phys A-Stat Mech Its Appl, 2019, 514: 786–795
2 Pei X, Pan Y, Wang H, et al. Empirical evidence and stability analysis of the linear car-following model with gamma-distributed memory effect. Phys A-Stat Mech Its Appl, 2016, 449: 311–323
3 Sipahi R, Niculescu S I. Stability of car following with human memory effects and automatic headway compensation. Proc R Soc A, 2010, 368: 4563–4583
4 Treiber M, Helbing D. Memory effects in microscopic traffic models and wide scattering in flow-density data. Phys Rev E, 2003, 68: 046119
5 Ammoun S, Nashashibi F. Real time trajectory prediction for collision risk estimation between vehicles. In: Proceedings of IEEE 5th International Conference on Intelligent Computer Communication and Processing, 2019. 417–422
6 Hillenbrand J, Spieker A M, Kroschel K. A multilevel collision mitigation approach—its situation assessment, decision making, and performance tradeoffs. IEEE Trans Intell Transport Syst, 2006, 7: 528–540
7 Batz T, Watson K, Beyerer J. Recognition of dangerous situations within a cooperative group of vehicles. In: Proceedings of IEEE Intelligent Vehicles Symposium, 2009. 907–912
8 Lefévre S, Vasquez D, Laugier C. A survey on motion prediction and risk assessment for intelligent vehicles. Robomech J, 2014. doi: 10.1186/s40648-014-0001-z
9 Tay M K C, Laugier C. Modelling smooth paths using gaussian processes. In: Field and Service Robotics. Berlin: Springer, 2008. 381–390
10 Tran Q, Firl J. Online maneuver recognition and multimodal trajectory prediction for intersection assistance using non-parametric regression. In: Proceedings of IEEE Intelligent Vehicles Symposium Proceedings, 2014. 918–923
11 Aoude G, Joseph J, Roy N, et al. Mobile agent trajectory prediction using Bayesian nonparametric reachability trees. In: Proceedings of American Institute of Aeronautics and Astronautics, 2011. 1512–1529
12 Lee J D, Moeckli J, Brown T L, et al. Distraction Detection and Mitigation Through Driver Feedback. National Highway Traffic Safety Administration, 2013
13 Zheng Y, Sathyanarayana A, Hansen J. Non-Uniform Time Window Processing of In-vehicle Signals for Maneuvers Recognition and Route Recovery. SAE Technical Paper, 2015
14 Li G, Li S E, Liao Y, et al. Lane change maneuver recognition via vehicle state and driver operation signals—results from naturalistic driving data. In: Proceedings of IEEE Intelligent Vehicles Symposium, 2015. 865–870
15 Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput, 1997, 9: 1735–1780
16 Sak H, Senior A, Beaufays F. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In: Proceedings of the 15th Annual Conference of the International Speech Communication Association, 2014
17 Xing Y, Lv C, Cao D. Personalized vehicle trajectory prediction based on joint time-series modeling for connected vehicles. IEEE Trans Veh Technol, 2020, 69: 1341–1352
18 Altché F, de La Fortelle A. An LSTM network for highway trajectory prediction. In: Proceedings of IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), 2017. 353–359
19 Kim B, Kang C M, Kim J, et al. Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network. In: Proceedings of IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), 2017. 399–404
20 Alahi A, Goel K, Ramanathan V, et al. Social LSTM: human trajectory prediction in crowded spaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016. 961–971
21 Deo N, Trivedi M M. Multi-modal trajectory prediction of surrounding vehicles with maneuver based LSTMs. In: Proceedings of IEEE Intelligent Vehicles Symposium (IV), 2018. 1179–1184
22 Xing Y, Lv C. Dynamic state estimation for the advanced brake system of electric vehicles by using deep recurrent neural networks. IEEE Trans Ind Electron, 2019. doi: 10.1109/TIE.2019.2952807
23 Wang X, Jiang R, Li L, et al. Capturing car-following behaviors by deep learning. IEEE Trans Intell Transp Syst, 2018, 19: 910–920
24 Shen Y, Tan S, Sordoni A, et al. Ordered neurons: integrating tree structures into recurrent neural networks. 2018. ArXiv: 181009536
25 Dai S, Li L, Li Z. Modeling vehicle interactions via modified LSTM models for trajectory prediction. IEEE Access, 2019, 7: 38287–38296