33rd European Modeling & Simulation Symposium
18th International Multidisciplinary Modeling & Simulation Multiconference

ISSN 2724-0029 ISBN 978-88-85741-57-7 © 2021 The Authors.
doi: 10.46354/i3m.2021.emss.045

A LSTM-based method for simulation execution validity evaluation

YuLun Wang[1], Wei Li[1] and Jing Li[2,*]

[1] Control and Simulation Center, Harbin Institute of Technology, Harbin 150080, China
[2] Science and Technology on Complex System Control and Intelligent Agent Cooperation Laboratory, Beijing 100074, China
*Corresponding author. Email address: wyl19972021@163.com

Abstract

This paper improves the concept of simulation execution validity and a methodology to assess the complex simulation system. As the complexity of the simulation system gradually rises, the magnitude of the simulation data obtained continues to increase, which makes it extremely difficult for experts to provide expert knowledge to evaluate the execution validity of the simulation system. More precisely, the essence of execution validity is to dig out the hidden relationships in the simulation time series data and complete the classification task whether it is valid. Considering that machine learning can better complete the two tasks of mining data features and classification, this paper adopts long short-term memory, a neural network used to process time series data, to evaluate the execution validity. Finally, an experiment is conducted on a simulation system, and the results show that the evaluation method based on LSTM can accurately evaluate the validity of the simulation system, and can greatly improve the efficiency of evaluation.

Keywords: Simulation Evaluation; Execution Validity; Time Series Data; LSTM

1. Introduction

Simulation is an approximate imitation of system operation by building mathematical and physical models on the computer to reflect the operating behavior characteristics of the actual system. System simulation is mainly composed of the process of model definition, mathematical modeling, model conversion, experimental design, simulation execution, simulation evaluation, and simulation evaluation is an important part of it. The evaluation of the simulation system is an essential approach to check whether the design of simulation system is reasonable, and according to the different evaluation objects, it can be divided into the evaluation of the simulation model and the evaluation of the simulation execution process. First of all, using model credibility to evaluate the consistency between the actual system and the simulation system is a prerequisite for simulation and a key issue in the field of simulation. Secondly, in the simulation execution process, this paper supposes that in order to avoid the simulation results that are inconsistent with the simulation design and to ensure that the simulation execution process is accurate and valid, a general method should be proposed to evaluate the real-time simulation data.

Time series data refers to a collection of data that changes over time. When evaluating a simulation system, it is often faced with massive amounts of simulation data, which generally have more complex characteristics such as nonlinearity, aperiodicity, and irregularity. It is possible to predict what will happen in the future by analyzing and mining the changing laws of the data, recording the fluctuations of the time series data, and according to the historical simulation data already possessed. Time series data mainly has the characteristics of strong randomness, non-stationary sequence, multi-dimensionality, massiveness, periodicity, and trend. At present, the traditional methods of analyzing time series data have obvious limitations when faced with the problem of non-linear time series data and multi-dimensional time series data jointly affecting the predicted variables, and they have the disadvantage of poor prediction accuracy.

For the problems above, this paper defines the execution validity of simulation, and proposes a method of evaluating the execution validity based on LSTM. Section 2 of this paper gives the definition of execution validity. Section 3 summarizes the principle of LSTM and gives more details about this method. Section 4 evaluates the execution validity of a simulation process to verify the feasibility of the method.

2. Definition of Execution Validity

The execution validity of the simulation system is a measure of reliability based on real-time simulation data. It is investigated that the data obtained by the simulation operation at evert discrete time during the execution process of the simulation system can reflect the accuracy of the actual system. According to the degree of validity of the real-time simulation data obtained, this paper divides the evaluation results of execution validity into three categories, namely "completely valid", "partially valid", and "basically invalid". The definition of execution validity mainly includes three levels: one is a measure of validity, which qualitatively analyzes the degree of validity; the other is an evaluation index for execution time, which is different from some evaluation methods that are oriented to historical data and for the data of the entire time period of the simulation, the execution validity is a dynamic evaluation index; third, it reflects the consistency between the current simulation system execution process and the ideal system execution process. Be aware of the validity of the current simulation time can enable the simulation staff to take corresponding decisions such as stopping the simulation when the execution process is found to be invalid, which can shorten the simulation execution cycle and improve the simulation efficiency.

3. LSTM-based Execution Validity Evaluation Method

3.1. Data Resource and Preprocessing

The data used in this paper comes from a cooperative flight simulation process, which contains a total of 54 736 discrete time points and a total of 16 entity variables. Including aircraft 3D coordinates, 3D speed, Mach number and other data information, partly shown in table 1.

Table 1. Part of data set.

Time	V_x	V_y	...	t_p	label
0	-214.94	-14.48	...	347.62	completely valid
1	-208.61	0.87	...	346	completely valid
2	-215.86	1.99	...	351.46	partially valid
...	...	...	...	...	...
54735	-0.02	45.77	...	4054.39	basically invalid

Except that most of the data are floating-point numbers that can normally participate in calculations, some simulation entity variables are of string type, and the classification labels are also string labels of "completely valid ", "partially valid", and "basically invalid". It needs to be processed before being used as training set data and converted into numerical variables. At the same time, because different variables have different data dimensions and different magnitudes, if they are directly input to the neural network without processing, it will cause the disadvantages of slow gradient descent when optimizing network parameters. Normalizing the data to the same fixed interval can make variables with different dimensions and characteristics have the same transformation scale, and can make each variable have the same effect on gradient descent.

This paper uses the z-score method to preprocess floating-point numbers. The z-score method is the most common data standardization method, also called standard deviation standardization. This method is based on the mean and standard deviation of the original data to standardize the data. The specific formula Given by Equation (1), μ is the mean of all the values of the variable in the sample data, and σ is the standard deviation of all the values of the variable. x' is the new value after normalization of the data.

$$\begin{aligned} \bar{x} &= \frac{1}{n}\sum_{i=1}^{n} x_i \\ \sigma &= \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})^2} \\ x' &= \frac{x-\bar{x}}{\sigma} \end{aligned} \tag{1}$$

Next, the string type data is processed, using “LabelEncoder” method to number discontinuous digital variables or text. The main principle is to first calculate the number of all unique values in the variable value, and number them in the order of appearance (0, 1, 2...), and finally replace the values in the original sample data with Corresponding to the number, so that non-continuous or text variables can be converted into numerical variables that can be calculated. The “OneHot” encoding format is used for tags, which is also the basic format used to solve classification problems in machine learning.

3.2. LSTM Construction Process

In this paper, LSTM network is constructed to classify and predict the result of the execution validity. It specifically includes the input layer, the LSTM hidden layer, the Dropout layer, and the Softmax layer. The overall structure is shown in Figure 1. Then, each layer will be explained in detail.

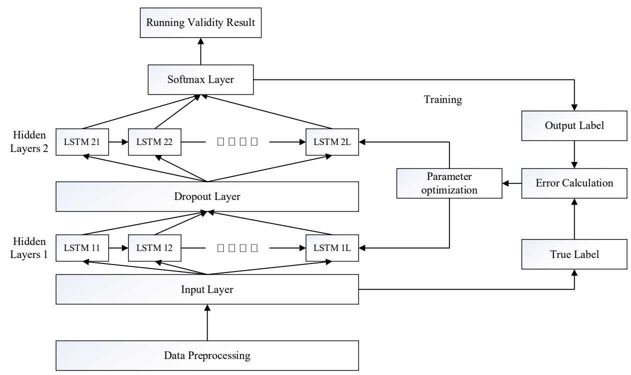


Figure 1. Overall structure of LSTM network

1. Input Layer.

The concept of time step is added to LSTM, and a sliding window is needed to process the training data. After the parameter is given, assuming timestep=k, each input sample contains data during k time step.

2. LSTM Layer.

LSTM inherits the advantages of recurrent neural network that can expand hidden layers in multiple layers and transmit information. It optimizes and improves the structure of each layer of neural network, and adds three kinds of "gate" settings, namely: forget gate, input gate, output gate. The forget gate is selectively forgetting. It selects how much incoming information was discarded at the last moment and how much incoming information can enter the current moment for calculation. The input gate consists of two parts: part of the information at the previous moment and the input information at the current moment. The output gate determines how much information can be output at the current moment. The specific structure is shown in Figure 2.

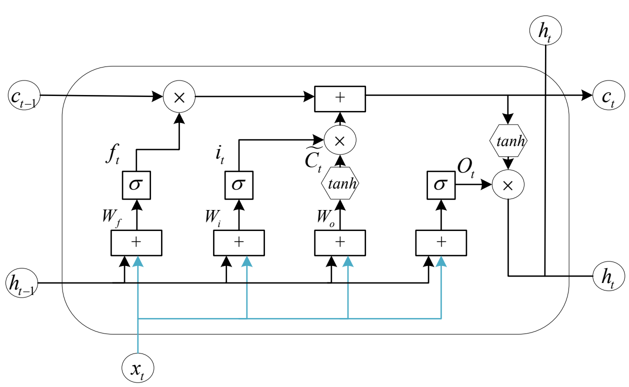


Figure 2. Structure of LSTM cell

The calculation process is given by equation (2)

$$
\begin{cases}
f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\
i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \\
c_t = f_t \otimes c_{t-1} + i_t \otimes \tilde{c}_t \\
o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\
h_t = o_t \otimes \tanh(c_t) \\
\sigma(x) = \dfrac{1}{1+e^{-x}} \\
\tanh(x) = \dfrac{\sinh(x)}{\cosh(x)} = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}
\end{cases} \tag{2}
$$

3. Dropout Layer

The Dropout layer randomly cuts some neurons in each training batch, that is, these neurons do not participate in information calculation and transmission. The clipped neurons do not participate in parameter optimization, and keep the previous value, while the remaining neurons continue to be updated according to the gradient descent method. When there are more neurons, the network structure is more complex, and the accuracy of the established network on the training set will be close to 100%, but the performance on the test set is not good, which is called overfitting in machine learning. That is, the model has poor ability on generalization. The reason why adding the dropout layer is to prevent the neural network from overfitting. Experiments show that the result is best when the dropout rate is equal to 0.5.

4. Softmax Layer

The Softmax layer is the most commonly used classifier in machine learning. It can calculate the probability of each label. The specific calculation method is given by equation (3).

$$
X = \{x_1, x_2, \dots x_n\}, \quad p\{label = i\} = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{3}
$$

3.3. Assessment Indexes

This paper uses Confusion Matrix and ROC curve to judge the accuracy of the classification task. The two indexes are briefly introduced below.

Confusion matrix can visualize actual and predicted results to show the accuracy of prediction, usually used in supervised learning. Each column is filled with the predicted result, and each row is filled with the actual label. This matrix can clearly see how many are accurately classified and how many are confused into other categories. Confusion matrix is used to analyze the accuracy of prediction to evaluate the model.

Table 2. Confusion matrix for binary classification.

Predict / Actual	Positive	Negative
True	TP	FN
False	FP	TN

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

The ROC curve shows the tradeoff between the true detections and false detections. Two calculation indicators are introduced into the ROC curve to draw the curve, namely TPR (True Positive Rate), FPR (False Positive Rate). In addition, we call the area under the ROC curve AUC, which is also a common parameter for evaluating the classification accuracy of the classifier.

4. Result and Discussion

Several experiments were conducted in the research to optimize the neural network model. The loss function uses cross entropy. In order to avoid overfitting, the dropout rate is set to 0.5. The sample data is divided into different training sets and test sets according to different sample sizes. The learning rate and batch data size during training are set to 0.001 and 149 respectively. The neural network is built using Tensorflow (version 1.15.0) and Keras (2.3.1).

First, the number of hidden neurons in the two LSTM layers' unit is 128, the time step is selected as 1, and the number of iterations is set as 300. Since the time step is selected to be 1, then the data of every time's input layer only contains one moment, which is equivalent to not reflecting the feature that LSTM can mine the implicit relationship between several steps of time series data, that is to say, LSTM degenerate into a general feedforward neural network in this experiment. Use such a setting to compare with the training results of a large time step later. The confusion matrix and ROC curve of training and testing process are shown in Figure 3 and Figure 4.

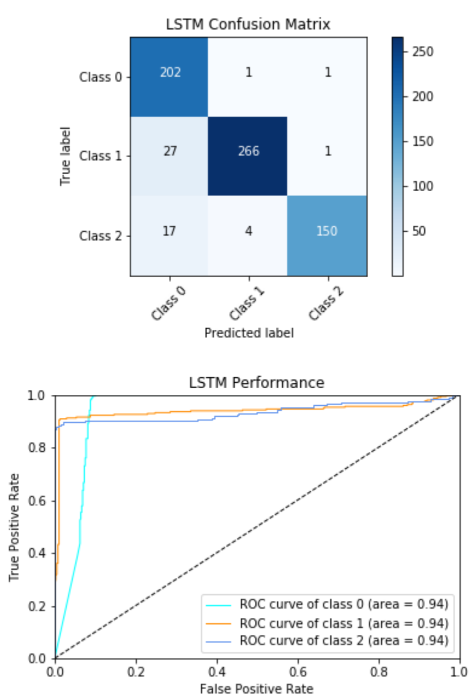


Figure 3. Confusion matrix and ROC curve of training when timestep equals 1

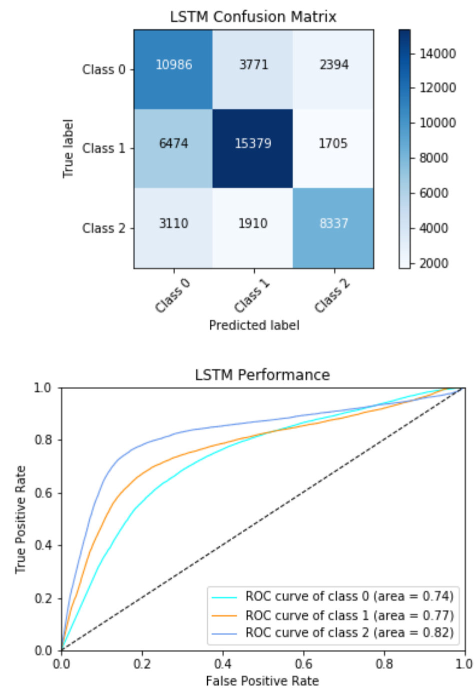
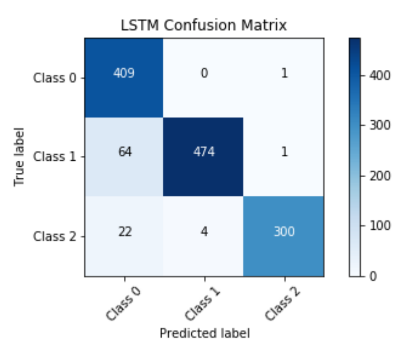


Figure 4. Confusion matrix and ROC curve of testing when timestep equals 1

Through the confusion matrix and ROC curve, it can be seen that the performance of the training set and the test set need to be improved. Through calculation, the discrimination accuracy in the confusion matrix is 64.18%, the AUC values of the three categories are 0.74,0.77,0.82, which is also relatively low. Next, the time step is selected as 30, and the other parameters are unchanged, retraining the model. Figure 5 and Figure 6 shows the confusion matrix and ROC curve of this training and testing process. It is proved that the use of LSTM for modeling can complete the classification task better than the feedforward neural network, and it can also better evaluate the execution validity of the simulation.



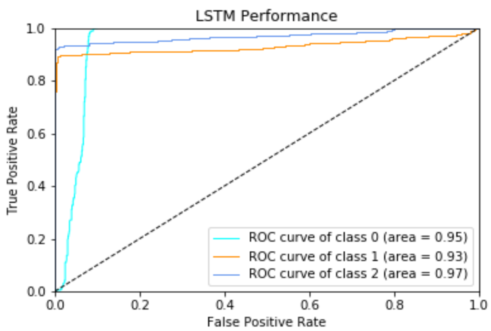
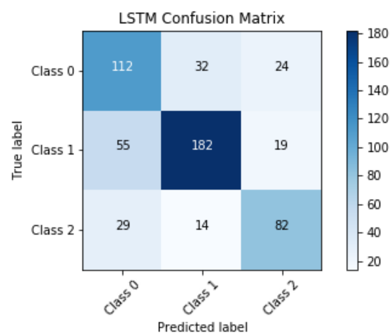


Figure 5. Confusion matrix and ROC curve of training when timestep equals 30



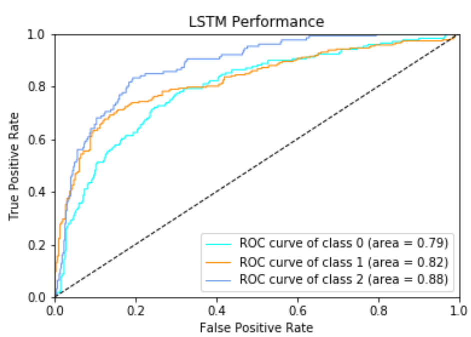
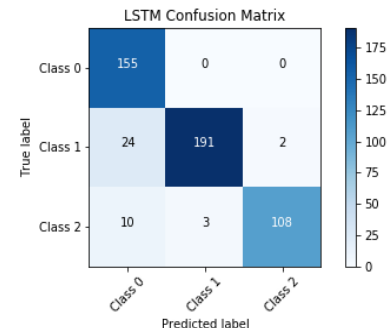


Figure 6. Confusion matrix and ROC curve of testing when timestep equals 30

Through the confusion matrix and ROC curve, it can be seen that the performance of the training set and the test set need to be improved. Through calculation, the discrimination accuracy in the confusion matrix is 68.49%, the AUC values of the three categories are 0.79,0.82,0.88, which is nearly 10% improvement compared to the last experiment. This also shows that LSTM can effectively mine the implicit relationship between time series data for prediction. At last, the time step is selected as 90, and still remain other parameters, retraining the model. Figure 7 and Figure 8 shows the confusion matrix and ROC curve of this training and testing process.



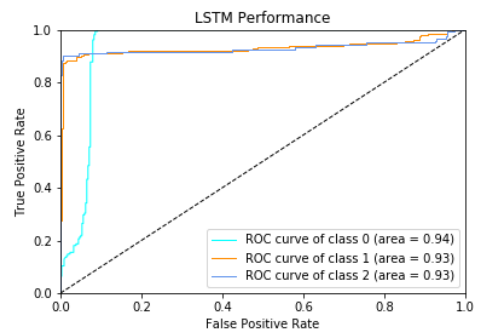
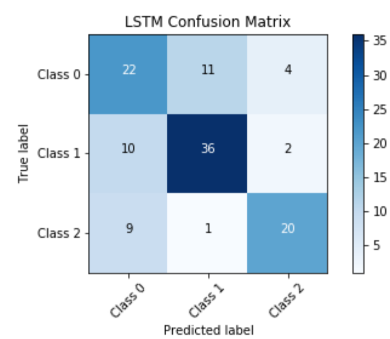


Figure 7. Confusion matrix and ROC curve of training when timestep equals 90



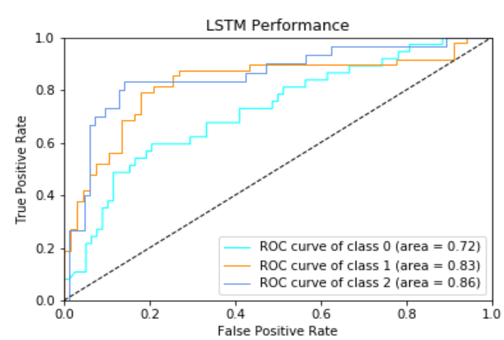


Figure 8. Confusion matrix and ROC curve of testing when timestep equals 90

Through the comparison of the calculation of confusion matrix and ROC curve, it can be seen that the prediction accuracy of the confusion matrix is about 67.83%, which is close to the result when timestep equals 30. But the AUC values of every category are less than that in the second experiment.

This paper supposes that there may be two explanations for such result. One is due to the limited training set data. During the expansion of the time step, the number of training samples will continue to decrease, and the fitting ability of the model may be poor, resulting in poor prediction accuracy on testing set. It is also possible that the time step is selected too large, and there may be too much information to be mined, which leads to overfitting and a relative decrease in the prediction result on testing set. In the actual training process, it is necessary to ensure that the sample data used for learning is sufficient, and the simulation staff can optimize the time step size and other parameters according to the training results. When founded the model parameters with the strongest generalization ability , then establish the LSTM model and use it for the evaluation of execution validity of simulation.

Compared with the expert system, although the inferring process of the expert system can be processed in a multi-threaded program, however, numerous rules still make the evaluation time unable to meet the real-time requirements of execution validity evaluation. The evaluation method based on LSTM only needs to learn multiple times offline to find better model parameters, and then save the parameter of the model in a document with h5 suffix. Before evaluation, simply load that document and the evaluation method only involves matrix operations, which enables the simulation system execution in real time to quickly calculate evaluation results of execution validity and improve evaluation efficiency.

5. Conclusion

Aiming at the problem of not having a complete concept of execution validity in the simulation evaluation, this paper gives the definition of execution validity. Then this paper proposes a long short-term memory neural network to process time series data and evaluate the execution validity. Compared with previous studies, this method does not require manual feature extraction. Using the characteristics of LSTM, it can perform selective learning and forgetting from the input data to make predictions. This method can effectively shorten the evaluation time and has the potential to become an auxiliary evaluation tool for the evaluation of execution validity. In the LSTM-based evaluation method, the acquisition of historical sample data may be an urgent problem to be solved. The scale of the sample data has a significant impact on the LSTM training model. How to obtain better sample data is a problem that can be studied in the next step.

Funding

This research is supported by the National Key R&D Program of China(Granted No. 2018YFB1701603)

References

Boer, P. , Kroese, D. P. , Mannor, S. , & Rubinstein, R. Y. . (2005). A tutorial on the cross-entropy method. Annals of Operations Research, 134(1), 19-67.

Elman, J. L. . (1990). Finding structure in time. Cognitive Science, 14(2), 179-211.

Kingma, D. , & Ba, J. . (2014). Adam: a method for stochastic optimization. Computer Science.

Lv, Z. , H Ding, Wang, L. , & Zou, Q. . (2021). A convolutional neural network using dinucleotide one-hot encoder for identifying dna n6-methyladenine sites in the rice genome. Neurocomputing, 422(4), 214-221.

Reynolds, M. S. . (2010). Dynamically reconfigurable antennas for rfid label encoders/readers. US.

Sundermeyer, M. , R Schlüter, & Ney, H. . (2012). LSTM Neural Networks for Language Modeling. Interspeech.

Wahlbeck, K. , Tuunainen, A. , Ahokas, A. , & Leucht, S. . (2001). Dropout rates in randomised antipsychotic drug trials. Psychopharmacology, 155(3), 230-233.