Target Operator Trajectory Prediction Method Based on Attention Mechanism and LSTM

Tongfei Shang*, Kunpeng Xiao and Kun Han

College of Information and Communication, National University of Defense Technology, Xi’an, China;

*Corresponding author e-mail: xietao09@nudt.edu.cn

Abstract. Aiming at the problem of incomplete information in wargames, this paper uses the encoding-decoding model of Long Short-Term Memory (LSTM) to divide the target trajectory prediction into two decoupling processes, encoding and decoding, using the decoding process Output the parameters of the learning probability distribution for trajectory prediction, and introduce an attention mechanism. Two attention layers are added to the LSTM model and the realization principle of the attention layer is introduced in detail. The context information is enriched to achieve the improvement of the model, and finally the simulation analysis The effectiveness of this method is verified.

Keywords: Attention Mechanism, Long Short-Term Memory, Operator Trajectory Prediction

1. Introduction

The commander will evaluate the surrounding environment and make judgments in wargame, so as to formulate his own feasible route. For our side, the enemy's initial deployment of troops is known. After continuous deduction, some of the enemy's operators have escaped our visibility through concealment and marching, resulting in a situation of incomplete information. In this paper, the target operator trajectory prediction is regarded as a sequence-to-sequence problem, and the time sequence of all operator trajectories in historical data is learned through the deep learning method to complete the prediction of the operator trajectory in the future. [1, 2, 3]

2. Target Operator Prediction Model Based on LSTM

In this paper, an LSTM model is established for each target operator that needs to be predicted, and the prediction of the operator position is divided into two decoupling processes: LSTM is used to mine the observed sequence information in the encoding process, and LSTM is used in the decoding process. The most likely operator position is output based on the prediction obtained from the encoding process. The entire model consists of an encoder (Encoder) and a decoder (Decoder), and the two are connected by a context vector. [4, 5, 6]

2.1 LSTM Encoder

...
During the encoding process, two inputs are received at each moment: the hidden layer representation of the operator position at the previous time and the position coordinates of the operator input at the current time. The hidden layer representation is as shown in formula (1):

$$h_t = f(h_{t-1}, X_t)$$  
(1)

In the formula, $f$ represents a cyclic unit in LSTM. Through the observation sequence input at each moment, after calculating the hidden layer representation at each moment, the hidden layer information is summarized to generate the context vector $C$. In the structural design of LSTM, after the calculation at the current moment is completed, directly replace $h_t$ with $C$, and the entire coding process is shown in the formula (2):

$$C = \text{LSTM}(X_1, X_2, ..., X_{obs})$$  
(2)

The encoding process is to map the observable sequence to $C$, which $C$ will be regarded as the information medium from the observation sequence to the prediction sequence. Combined with the feature extraction ability of LSTM on the observation sequence, the encoding process obtains the representation of the hidden layer through learning, which provides effective information for the subsequent decoding process. [7, 8]

### 2.2 LSTM Decoder

In the decoding process, two inputs are received at each moment: the context vector $C$ and the predicted position sequence, and the possible position of the next moment of the operator after nonlinear transformation is output. The whole process is shown in formula (3):

$$Y_t = f(C), Y_2 = f(C, Y_1), Y_3 = f(C, Y_1, Y_2), ..., Y_{i+1} = f(C, Y_1, Y_2, ..., Y_t)$$  
(3)

Among them, $f$ represents a cyclic unit in the LSTM decoding process.

The essence of this process is to decompose the joint probability of the output sequence into sequential conditional probabilities: as shown in formula (4):

$$P(Y) = \prod_{t=1}^{T} P(Y_t | \{Y_1, Y_2, ..., Y_{t-1}\}, C)$$  
(4)

Due to the interconnection at each moment, the relationship between the output of the decoding process and the hidden layer can be expressed as a conditional probability, as shown in formula (5):

$$P(Y_t | \{Y_1, Y_2, ..., Y_{t-1}\}) = g(Y_{t-1}, s_{t-1}, C)$$  
(5)

Among them, represents the hidden layer at the previous moment in the LSTM, is the context vector, is the output at the previous moment in the decoding process, and represents a loop unit in the LSTM decoding process after adding the function to calculate the output position probability. In the decoding process, when predicting the position of the operator at each moment, the context information of the entire observation sequence can be obtained according to the vector, and combined with the deep learning ability of LSTM, the non-interference between the prediction of the operator position at the next moment and the previous two can be established. Linear mapping relationship, complete the final output. [9, 10]

### 2.3 Location Prediction

In the process of wargaming, commanders make decisions in most cases according to conventional methods. The model cannot predict the position of the operator exactly at each moment, so point estimation is of no value. However, the model can estimate the probability distribution that the predicted position satisfies, and use this distribution as a hypothesis bias for model induction and
learning. According to the literature reference, suppose the operator appears The probability of a certain coordinate position obeys the two-dimensional Gaussian distribution, and the probability distribution of the predicted coordinate at a time \( t \) is shown in formula (6):

\[
(\hat{x}, \hat{y}) = N(\mu_i', \sigma_i', \rho_i')
\]  

(6)

In the decoding process, the hidden state \( s_{t-1}' \) at \( t-1 \) time is learned and given five parameters in the two-dimensional Gaussian distribution: expectation \( \mu_i' = (\mu_s, \mu_y)' \), standard deviation \( \sigma_i' = (\sigma_s, \sigma_y)' \), and correlation coefficient \( \rho_i' \). Train these five parameters by establishing a linear layer of the weight matrix \( 5 \times D \), as shown in formula (7):

\[
[\mu_i', \sigma_i', \rho_i'] = W_p s_{t-1}'
\]

(7)

In the formula, \( D \) represents the dimension of the hidden layer, and \( W_p \) represents the weight matrix of the predicted probability. The probability density function is shown in the formula (8) and (9):

\[
f(x, y) = \frac{1}{2\pi\sigma_s\sigma_y\sqrt{1-\rho^2}} \exp\left(-\frac{z}{1-\rho^2}\right)
\]

(8)

\[
z = \frac{(x-\mu_x)^2}{\sigma_x^2} + \frac{(y-\mu_y)^2}{\sigma_y^2} + \frac{2\rho(x-\mu_x)(y-\mu_y)}{\sigma_x\sigma_y}
\]

(9)

In the formula, \( \mu_x, \mu_y \) represents the predicted coordinates, that is, the estimated mean value of the Gaussian distribution. The probability density function of the two-dimensional Gaussian distribution can reflect the learning effect of the model. The higher the value, the more accurate the prediction. It should be noted here that the value of the probability density function \( f \) is not the probability of the predicted coordinate, and its integral is the probability of the predicted coordinate appearing in this area.

The training of the model needs to minimize the loss of all positions on the training set, because the probability density value is relatively small, and the logarithm is used to calculate the probability. The loss function is shown in formula (10):

\[
L'(W_p) = -\sum_{t=1}^{k} \log \left( P(s_t', y'_t | \sigma_i', \mu_i', \rho_i') \right)
\]

(10)

In the formula, \( L' \) represents the trajectory loss of the operator. Designing the loss function in this way makes the probability calculation change from multiplication to addition, avoiding the calculation of too small a value, and the probability of the true value is 1, the function itself is not affected by the multiplication operation, making the calculation more convenient. In the prediction probability distribution model, after each parameter is optimized and approximated by the above-mentioned minimized cross-entropy loss function, the training is completed.

3. Attention Mechanism

The processing process of introducing the attention mechanism is as follows: the hidden layer information \( h_i \ (i = 1, 2, ..., k) \) in the encoding process is weighted and averaged, as shown in the formula (11):
Among them: \( a_i \) represents the weight \( h_i \) of the corresponding.

After introducing the attention mechanism, training can be used to learn the influence weights of the predicted output at each moment in the decoding process on each hidden state in the encoding process, and thereby calculate the context vector that changes with time. In this way, in the decoding process, different correlation degrees will have different effects on the prediction results, and the prediction results are generated through iteration, as shown in formula (12):

\[
Y_i = f(C_i), Y_2 = f(C_2,Y_i), Y_3 = f(C_3,Y_i,Y_2),..., Y_{i+1} = f(C_i,Y_i,Y_2,...,Y_i)
\] (12)

Where \( C_i \) is the hidden layer sequence information \( h_j \) generated by the observation sequence \( X \) through the encoding process, and is obtained after weighted summation according to the normalized probability. The calculation process of \( C_i \) is shown in formula (13):

\[
C_i = \sum_{j=1}^{T_X} a_i h_j
\] (13)

In the formula, \( T_X \) is the length of the observation sequence, \( h_j \) is the hidden layer sequence obtained in the encoding process, and \( a_i \) is the weight.

The introduction of the entire attention mechanism is shown in Fig 1.

![Fig 1. Attention model for single operator trajectory prediction](image)

**4. Improved Model Based on Attention Mechanism and LSTM**

The overall process of the improved model is: the historical trajectory \( X = \{(x_1',y_1'),(x_2',y_2'),..., (x_{obs}',y_{obs}')\} \) of the first operator that can be observed in the early stage is processed by the embedding layer embedding and then input to the decoding layer. After processing, the hidden layer information \( h'_1, h'_2,.., h'_{obs} \) at each moment is obtained. Then through the attention layer, the hidden information sequence of the coding layer is weighted and summed to get the context information \( C'_i \) about the time dimension. As shown in formula (14), each moment \( C'_i \) is dynamically changing with time.

\[
C'_i = \sum_{j=1}^{T_X} a'_i h'_j
\] (14)

Where \( T_X \) is the length of the observable operator sequence, \( T_y \) is the length of the operator's predicted position sequence, \( h_j \) represents the hidden layer information of the operator at the first moment.
The derivation of the elements in the matrix is shown in formula (15) and (16):

\[ a_{ij} = \frac{\exp(E_{ij})}{\sum_{i=1}^{h} E_{ij}} \]  

(15)

\[ E_{ij} = A(s_{i-1}, h_{j}) \]  

(16)

Where \( E_{ij} \) is the scoring function mentioned earlier, through the normalization of the function, the probability value \( a_{ij} \) is obtained.

As a spatially related attention vector \( H_{i} \), its representation is shown in (17):

\[ H_{i} = \sum_{j=1}^{N} q_{ij} C_{ij} \]  

(17)

\[ q_{ij} = \frac{\exp(B(e_{i}, e_{j}; W_{i}, b_{j}))}{\sum_{k=1}^{N} \exp(B(e_{i}, e_{j}; W_{i}, b_{j}))} \]  

(18)

Where is the total number of operators and is the result of the observable input sequence calculated by the embedding layer. Establish another feedforward neural network as a scoring function.

5. Simulation Analysis

Using three data sets, including "mountain passages, urban residential areas, island terraces" encounter battle scenarios. The data set contains three scenarios of game data, each with 100 rounds, each with 20 rounds, a total of 48 operators of various types (16 in each scenario), a total of 96,000 operator trajectories, according to 4:1 The ratio is divided into training set and test set.

As the batch size increases, the time overhead decreases, especially as the batch size increases from 32 to 64. However, when the batch size is increased from 64 to 128, the speed does not improve greatly, and the error is also increased relative to the model with a batch size of 32, which shows that blindly increasing the batch size may not necessarily improve the performance of the model. The improved model of a layer of spatial attention mechanism (purple in the figure) is selected for comparison with the former two, and the experimental results are shown in Figure 2. It can be seen from the results that, even with the use of spatial correlation features between sub-substances, the performance of the attention mechanism is better than that of Social-LSTM. Comparing the performance of the two-layer attention mechanism improvement model (green in the figure) in the three rounds of experiments, the error in the third round is relatively smaller. This is mainly due to the fact that the island platform is centered, the field of view is relatively open, and there are more observable data. , The prediction difficulty is relatively small.

![Figure 2](image)

**Fig 2.** Comparison of the results of the scenario of the platform on the island

Taking the observation sequences of different lengths of "urban residential areas" as input, the prediction results are shown in Figure 3. From a horizontal perspective, the prediction result with an observation length of 15 (Observation-15) is more accurate than the previous two; from a vertical
perspective, the previous prediction is more accurate, especially when the observation sequence length is 15, the last 5 The predicted position of the round is not much different from the real position.

![Graph showing predicted position over time](image)

**Fig. 3** Comparison of prediction results under different observation step conditions

6. Conclusion

Target trajectory prediction is an important method to solve the problem of incomplete information in wargaming. Accurate prediction can provide more reliable information for subsequent decision-making. This article mainly introduces the improved LSTM model based on the attention mechanism, and proves its effectiveness through experiments. The feasibility of the trajectory prediction problem plays a supporting role for the subsequent decision-making model.

References

[1] Silver D, Huang A, Maddison C J, et al. Mastering the game of Go with deep neural networks and tree search. Nature, 2016, 529(7587):484-489.

[2] Silver D, Schrittwieser J, Simonyan K, et al. Mastering the game of Go with out human knowledge. Nature, 2017, 550(7676):354-359.

[3] Ceranowicz A. Modular Semi-automated Forces// Sadowski D A, Seila A F, Manivannan M S, et al. In Proceedings of the 26th Conference on Winter Simulation. Lake Buena Vista, FL, USA, 1994: 755–761.

[4] Binson K, Marsden C, McKenzie F, et al. Knowledge-Based Tactical Decision Making in the CCTT SAF Prototype. In Proceedings of The Fourth Conference on Computer Generated Forces and Behavioral Representation. Orlando, FL, USA, 1994.

[5] Giampapa J A, Sycara K P, Owens S, et al. Extending the One SAF Testbed into a C4ISR Testbed. Simulation. 2004, 80 (12): 681–691.

[6] Laird J E, Johnson W L, Jones R M, et al. Simulated Intelligent Forces for Air: The Soar/IFOR Project 1995. In Proceedings of the Fifth Conference on Computer Generated Forces and Behavioral Representation. Orlando, FL, 1995.

[7] Salisbury M R, Seidel D W, Booker L B. A Brief Review of the Command Forces(CFOR) Program // Lilegdon W R, Goldsman D, Alexopoulos C, et al. In Proceedings of the 27th conference on Winter simulation. Arlington, VA, USA, 1995:626–633.

[8] Gratch J. Task-decomposition Planning for Command Decision Making. In Proceedings of the Sixth Conference on Computer Generated Forces and Behavioral Representation, 1996.

[9] Ji Y, Massanari R M, Ager J, et al. A Fuzzy Logic-based Computational Recognition-primed Decision Model. Information Sciences. 2007, 177 (20):4338–4353.

[10] Sokolowski J A. Enhanced Decision Modeling Using Multi-agent System Simulation. Simulation. 2003, 79 (4): 232–242.