Human Trajectory Prediction Using Transformer and LSTM Models

Bingcheng Chen, Arvin Rokni
(Dated: October 2023)

Abstract Briefly explain the topic and summarize your findings. Should not be longer than 300 words, and for your projects probably shorter than 200 words.

I. INTRODUCTION

Predicting how people move is vital for autonomous systems in various fields like self-driving cars, video surveillance, and tracking objects. It's not just a technological achievement; it's crucial for safety and efficiency in our increasingly automated world. Human trajectory prediction is a challenging task because it involves forecasting the future paths of individuals, and human movement can be highly dynamic, complex, and influenced by a wide range of factors. In this paper, we explore a unique idea: using Transformer and Long short-term memory (LSTM) networks to predict how people will move. Both Transformer and LSTM models offer solutions to these challenges through their ability to capture long-term dependencies, model complex interactions, and adapt to various trajectory patterns, making them valuable tools in this field.

Inspired by the research paper called "Transformer Networks for Trajectory Forecasting," our goal is to use a type of LSMT and Transformer network to predict where people will go in a dynamic scene. By using LSTMs and Transformers for this, we aim to improve how we predict movement and make autonomous vehicles safer by avoiding accidents and planning better routes.

Human trajectories are influenced by spatial and temporal dependencies, making prediction challenging. Transformers can capture long-range dependencies and model complex relationships between past and future positions through their self-attention mechanisms while LSTMs are designed to capture temporal dependencies in sequential data. They can model complex temporal relationships within trajectory data, making them suitable for handling the temporal aspect of the prediction.

II. BACKGROUND THEORY

Trajectory prediction has gained many attention in recent years following the introduction of the paper "Attention is All You Need" by Vaswani et al. in 2017. Prior to the emergence of the transformer architecture, LSTM models were widely regarded as the top performers, owing to their suitability for time-series sequence problems. However, with the advent of transformer, researchers has shifted their attention to exploiting the power of transformer for trajectory prediction.

First, LSTM is the type of recurrent neural network architecture which is able to capture and remember long-term information, overcoming the vanishing gradients issue in typical RNN.

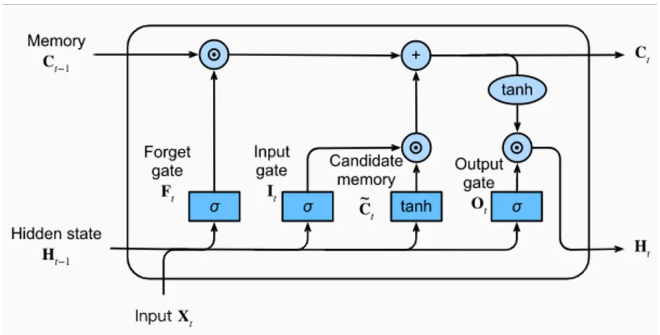

FIG. 1: LSTM Cell

$$
\begin{aligned}
i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\
g_t &= \tanh(W_g \cdot [h_{t-1}, x_t] + b_g) \\
o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\
c_t &= f_t \cdot c_{t-1} + i_t \cdot g_t \\
h_t &= o_t \cdot \tanh(c_t)
\end{aligned}
$$

Second, transformer architecture is characterized by its self-attention mechanism, which allows it to capture relationships between tokens in a sequence.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \qquad (1)$$

III. METHOD

In this paper, we explored three main related architecture of models: the first has focus on classical LSTM model. the second was encoder-decoder transformer, and the third was encoder-decoder LSTM with multihead attention.

A. LSTM Model (Many-To-Many)

As illustrated in Figure 2, we first implemented the classical LSTM architecture from scratch, we set hidden size

to 128 and number of hidden layers to 3. Additionally, the input sequence length is 8 and output sequence length is 12, we stacked all outputs from the last hidden layer and feeded into a two layer of Linear network.
This section describes your solution to the problem in detail.This section describes your solution to the problem in detail.This section describes your solution to the problem in detail.This section describes your solution to the problem in detail.This section describes your solution to the problem in detail.This section describes your solution to the problem in detail.This section describes your solution to the problem in detail.
Loss function for LSTM based models:

$$\text{Loss} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

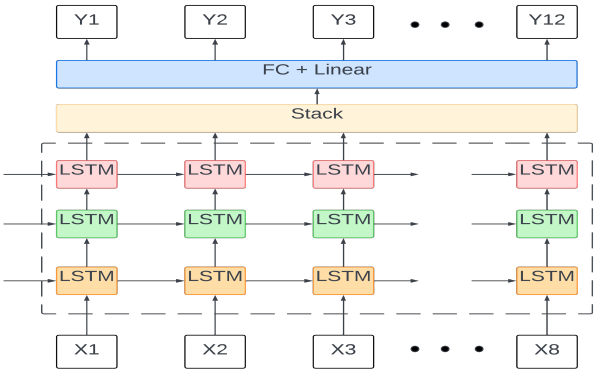

FIG. 2: LSTM (many-to-many)

B. Encoder-decoder Transformer

As is shown in Figure 3, the transformer has a encoder and decoder structure, the encoder processes input data and the decoder generates the output data. Compared with the classical transformer architecture, we add one more norm layer to both the encoder and decoder before the data is feed into the multi-head self attention layer, expecting an improved generalization ability to unseen data.
We also change the activation function to GELU [1] activation, which by its probability under a Gaussian distribution, instead of RELU activation gates the input by its sign. It combines a scaled linear component $(0.5x)$ with a scaled hyperbolic tangent component $(1 + \tanh(\sqrt{\frac{2}{\pi}}(x + 0.044715x^3)))$. This combination can help to capture a wider range of input values and can mitigate the vanishing gradient problem.

$$\text{GELU}(x) = 0.5x\left(1 + \tanh\left(\sqrt{\frac{2}{\pi}}(x + 0.044715x^3)\right)\right) \quad (2)$$

Compared with Sharon's work, we also removing the dropout layers considering that in this specific case that the dataset is not big and the model has generalized well, this has been confirmed by the performance on the validation data.

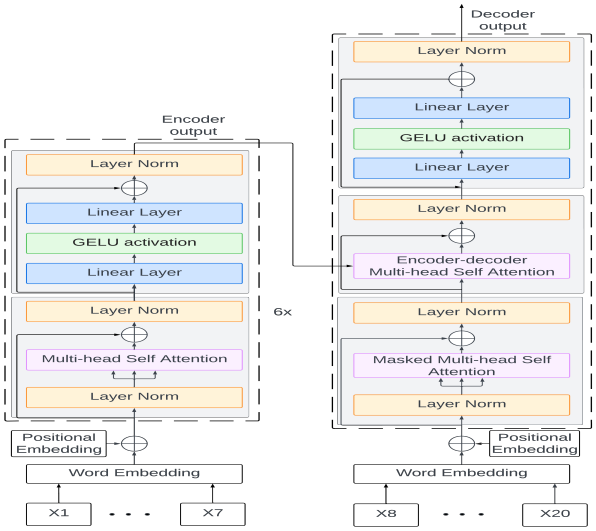

FIG. 3: Encoder-decoder Transformer

C. Encoder-decoder LSTM With Multihead Attention

In this section, we consider for trajectory prediction a third model which combine multi-head self attention with encoder-decoder LSTM architecture, as illustrated in Figure 4.

In the encoder part, an input multi-head attention (2 attention heads) is applied after the input sequences are processed by a word embedding layer (embedding size is 8) and norm layer. For each 'Encoder' in the figure it has 2 hidden layers, and we set hidden size to 128.

In the decoder part, another multi-head attention mechanism is used, the query comes from the output of the decoder in previous time step h^D_{t-1}, the keys and values comes from the corresponding output of the encoder $\tilde{h}^D_t$. we set the 2 head attention, the output of the cross multi-head attention is then feed into the decoder, which has the same structure as encoder, to generate the output of decoder.

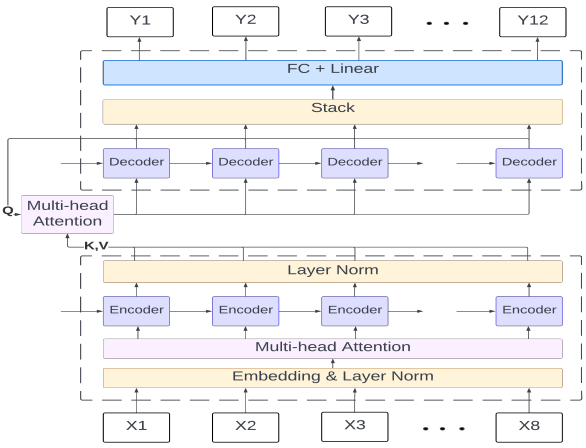

FIG. 4: Encoder-decoder LSTM With Multihead Attention

IV. RESULT

Same as the approach employed in other related papers, the evaluation of the model's performance is conducted through the use of two metrics: MAD and FAD. MAD, which stands for Mean Average Displacement, calculates the average discrepancy at each time step, providing insights into the model's overall performance throughout the entire time sequence prediction. On the other hand, FAD, or Final Average Displacement, specifically assesses the model's ability to capture the discrepancy at the final time step, evaluating on the models' effectiveness in predicting the final state.
Recalling the TrajNet dataset that we utilized, we are given 8 consecutive ground-truth values i.e., $t-7, t-6, \ldots, t$, the primary objective of the model is to predict the subsequent 12 values, spanning from $t+1$ to $t+12$. To compute the metrics MAD and FAD for each 12-value sequence prediction, the following formulas are applied:

$$\text{MAD} = \frac{1}{12}\sum_{t=1}^{12} \|y_t, \hat{y}_t\|$$

$$\text{FAD} = \sum_{t=12}^{12} \|y_t, \hat{y}_t\|$$

where, $\|y_t, \hat{y}_t\|$ is the euclidean distance between prediction and ground truth in time step t.
As illustrated in Figure 6
the transformer model performs best, followed by LSTM, with LSTM with Multihead Attention performing the least effectively.

Rank	Method	Avg	FAD	MAD	Cit	Year
2	*Transformer*	*0.725*	*0.998*	*0.452*		*2023*
3	*LSTM(Many-to-Many)*	*0.732*	*0.973*	*0.490*		*2023*
4	Transformer_sharon	0.767	1.059	0.474	[2]	2022
5	TF	0.776	1.197	0.356	[3]	2020
6	TFq	0.858	1.300	0.416	[3]	2020
7	*BERT*	0.879	1.354	0.440	[3]	2020
8	*BERT NLP pretrained*	0.902	1.357	0.447	[3]	2020
9	*LSTM With Attention*	*0.975*	*1.236*	*0.713*		*2023*

TABLE I: Results of different models, *Blue italic indicates methods implemented in this work.*

V. CONCLUSION

A short summary of your main results. It may also be a good idea to comment on possible future work.

The manuscript should include future directions of the research. Authors are strongly encouraged not to reference multiple figures or tables in the conclusion; these should be referenced in the body of the paper.

A short summary of your main results. It may also be a good idea to comment on possible future work.

The manuscript should include future directions of the research. Authors are strongly encouraged not to reference multiple figures or tables in the conclusion; these should be referenced in the body of the paper.

The manuscript should include future directions of the research. Authors are strongly encouraged not to reference multiple figures or tables in the conclusion; these should be referenced in the body of the paper.

A short summary of your main results. It may also be a good idea to comment on possible future work.

The manuscript should include future directions of the research. Authors are strongly encouraged not to reference multiple figures or tables in the conclusion; these should be referenced in the body of the paper.

The manuscript should include future directions of the research. Authors are strongly encouraged not to reference multiple figures or tables in the conclusion; these should be referenced in the body of the paper.

A short summary of your main results. It may also be a good idea to comment on possible future work.

The manuscript should include future directions of the research. Authors are strongly encouraged not to reference multiple figures or tables in the conclusion; these should be referenced in the body of the paper.

VI. REFERENCES

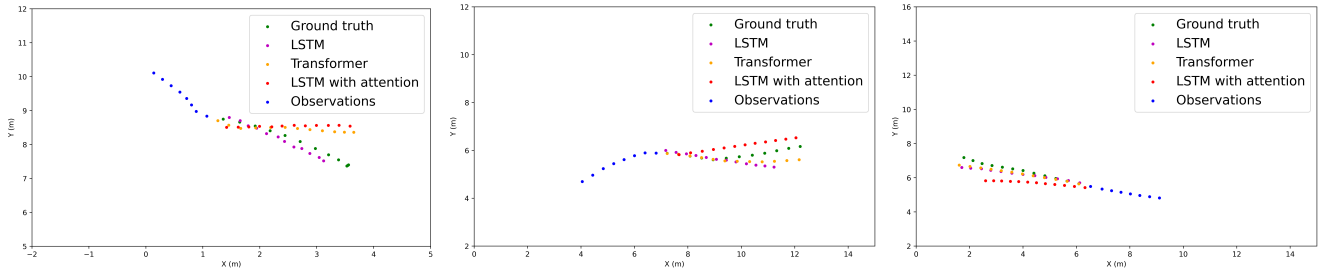

FIG. 5: Examples of Observation, Ground Truth and Predictions by three models

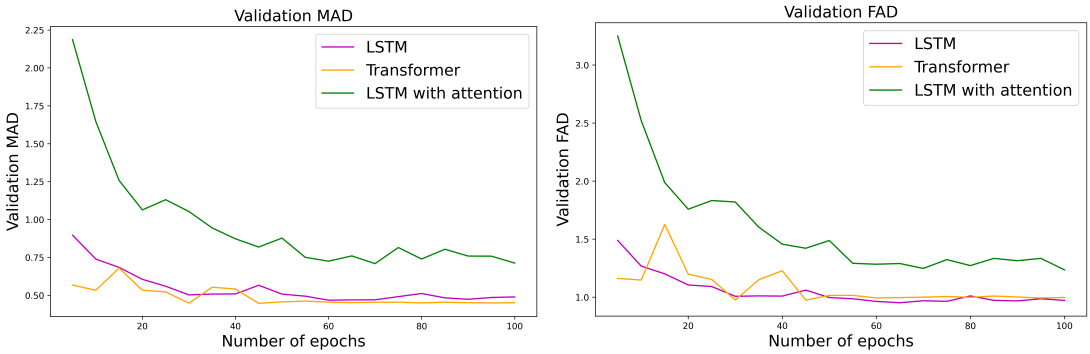

FIG. 6: MAD and FAD of three models

[1] D. Hendrycks and K. Gimpel, Gaussian error linear units (gelus) (2023), arXiv:1606.08415 [cs.LG].
[2] S. R. Shaji, Trajectory prediction transformers, `https://github.com/sharonrichushaji/trajectory-prediction-transformers` (2022).
[3] F. Giuliari, I. Hasan, M. Cristani, and F. Galasso, Transformer networks for trajectory forecasting (2020), arXiv:2003.08111 [cs.CV].