Final Report

Human Trajectory Prediction via Social LSTM

ABSTRACT

The project aims to reimplement Social LSTM: Human Trajectory Prediction in Crowded Space and study the difference between social LSTM and vanilla LSTM. The algorithms were tested on standard datasets from ETH and UCY.

The results show than RNN network was a valid strategy in terms of making predictions about human trajectory. Although Social LSTM did not quite outperform LSTM in terms of improving prediction accuracy, it did a better job avoiding collisions. The training datasets was preprocessed and scaled down 10 times intentionally to save the training time, which might sacrifice the accuracy. Also, the training curves did not fully converge when the training process was terminated.

Future improvements of the work would include considering training not only on spatial coordinates but also velocities, tailoring the outputs to control inputs of robots and taking into consideration of vision when calculating the social grid.

For codes and detail results, please see https://github.com/xywang0001/Social_LSTM.

1.Introduction

Prediction of human trajectory has been an important problem in the trajectory planning of social-aware robot. Vehicles navigating the scene would need to foresee the positions of pedestrians and adjust its own path. Researchers have investigated deep learning toolbox as an alternative for hand-craft functions to study the interactions and decisions of human. Social LSTM was first proposed in 2016 CVPR by Alahi etc., as a response to the progress of RNN network. It has soon gained considerable attention and became the foundation for a lot of strategies that came afterwards. The project is attempting to evaluate the method by reimplementing the network and testing the network on human trajectory dataset from ETH and UCY.

The problem is formulated as the following:

Knowing the position of all the targets from T_0 to T_{obs}, predict the trajectory from T_{obs} to T_{pred}.

In addition to vanilla LSTM, social LSTM implemented a social pooling layer that could consider interactions between the target and its neighbors. Also comparing to the quite famous social force model proposed by Helbing and Molnar [2], the social LSTM predict human interactions in a more data-driven fashion.

2. Method

2.1 Network Structure

The network uses a standard LSTM network. The code is tailored to provide a gru option.

The following is a summary of how the network is constructed. Vanilla LSTM removes all the social tensor and grid mask portion.

Class Social LSTM
Input: current positions, number of pedestrians in each frame, hidden states of the peds, cell states of the peds
Outputs: 2D Gaussian distribution parameters
Procedure:
Initialization *parameters: rnn_size, grid_size, embedding size, input_size, output_size, ifgru*

1. For each frame, do
2. Get pedID for pedestrians present in the current frame
3. Select the corresponding input positions from current positions (nodes)
4. Get the corresponding grid mask
5. Get the corresponding hidden and cell states
6. Compute the social tensor
7. Embedding input
8. Embedding social tensor
9. complete_input = em_input || em_social_tensor
10. 1 step LSTM cell run
11. Update hidden and cell states
12. Get output

2.2 Loss

The loss used during the training process is defined as the negative log-likely hood loss.

2.3 Implementation details

Following the paper, an embedding dimension of 64 was used for the spatial coordinates. The spatial pooling size was set to be 32. The fixed hidden dimension was set to be 128 for both vanilla LSTM models and Social LSTM models. The learning rate is set to be 0.001 and the optimizer is RMS-prop. The sequence length fed into the network is set to be 20, with 8 steps for observation and 12 for prediction.

ETH and UCY provide 5 datasets combined, with 2 from ETH and 3 from UCY. The project utilizes the leave-out strategy from the paper and trained the network on 4 datasets with 1 left for testing.

Also, to fasten the training speed, in the project the training data is loaded every ten frames from the raw data.

3. Results

3.1 Final Error

The performance was evaluated on how the network perform on the leave-out dataset.

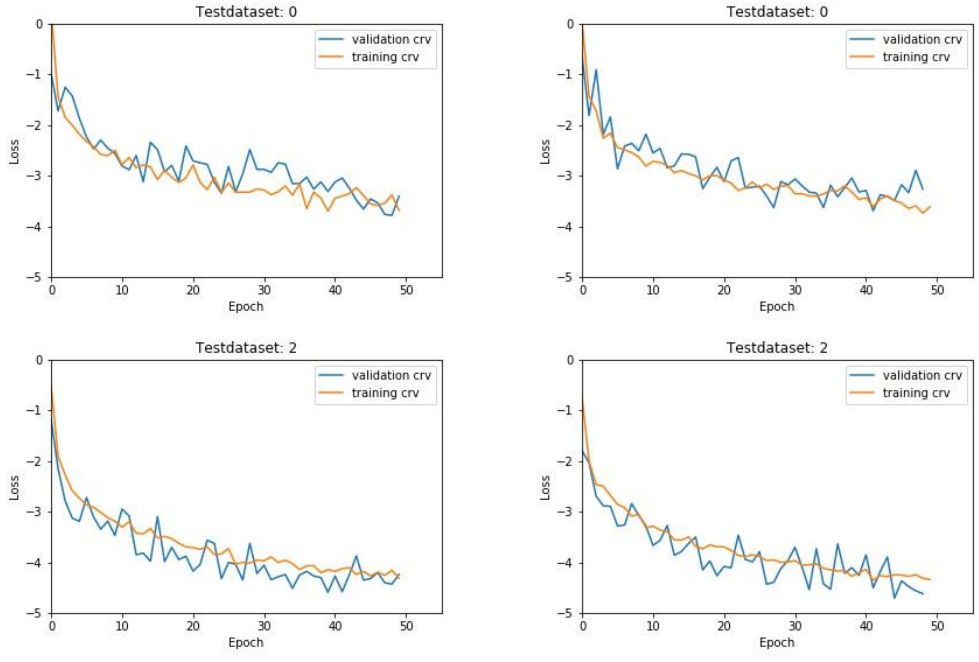
Table 1 Final Errors and Mean Error on test sets

Test set	V-LSTM		S-LSTM	
	Final err	Mean err	Final err	Mean err
0	0.74	0.081	0.85	0.090
1	0.68	0.080	0.66	0.070
2	0.11	0.040	0.08	0.033
3	0.74	0.078	0.62	0.078
4	1.02	0.073	0.78	0.069

3.2 Standard training curve& validation curve

The network was trained for 50 epochs with a relatively small batch size.

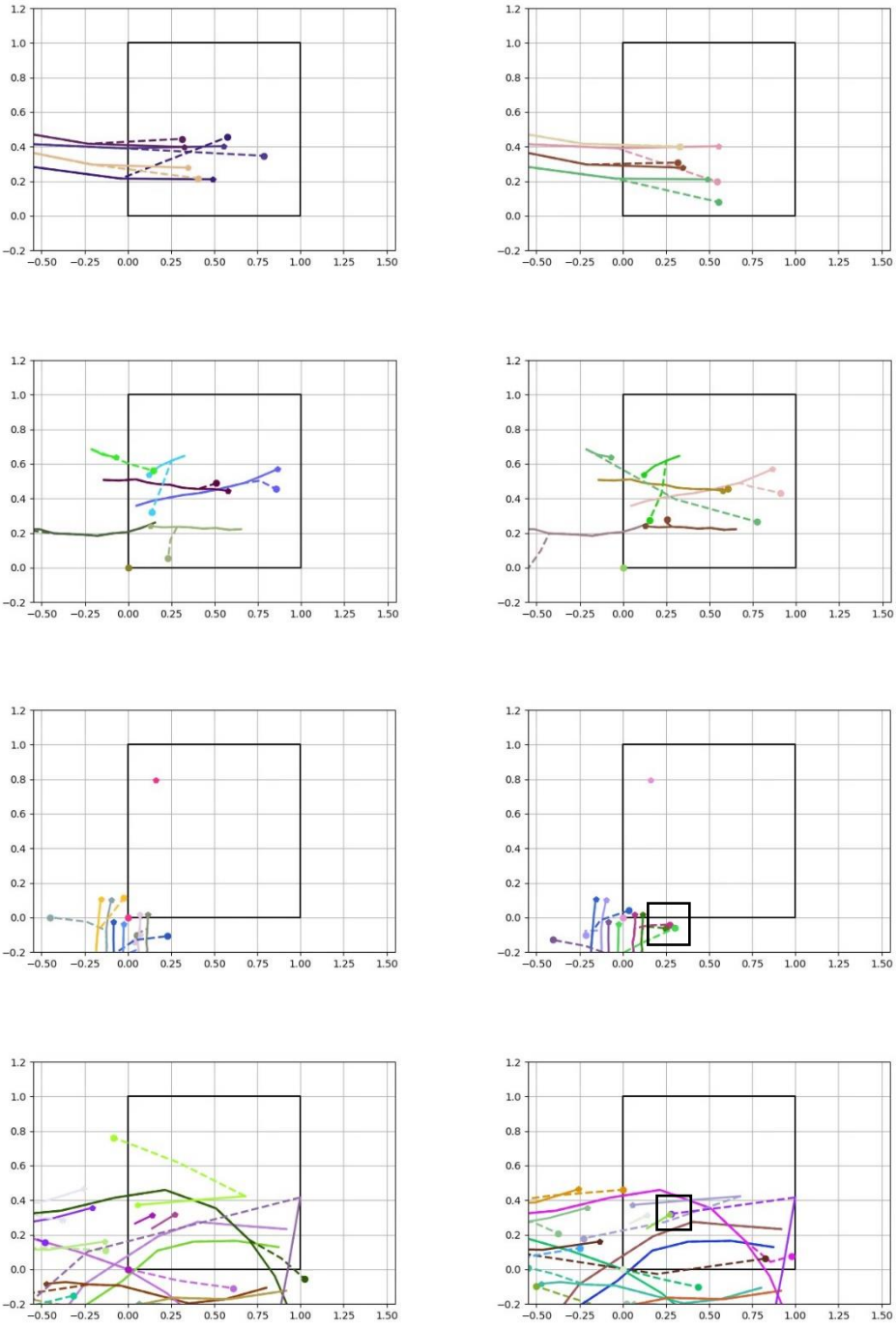
The following shows the loss during training and validation process on the dataset that has the best performance and the dataset with the worst performance. The difference could be caused by the different number of training samples in each dataset. In both cases the curves did not fully converge when the training process is terminated. However, there were not obvious difference between social LSTM and vanilla LSTM.



Training with social LSTM *Training with vanilla LSTM*

Figure 1 Training Curve and Test Curve

3.3 Results visualization



Social LSTM *Vanilla LSTM*

Figure 2 Result Visualizations

Here are some selected scenarios when testing on different datasets. As shown in the figures, the prediction accuracy on both models are still quite low. Some predicted movements have also been quite aggressive due to a lack of velocity control. Social LSTM did not show significant improvements in terms of prediction accuracy. However, it did a better job avoiding collisions. There were some collisions happening in the last two scenarios for Vanilla LSTM.

4. Future work

The training output is 2D gaussian parameters that could generation the coordinates of the next position. To apply the system to social aware robots. It could be more constant with the state-of-art theory to have the output as control parameters that could be fed to robot.[3]

The social pooling layer implemented did not take into consideration of the vision. It would take into consideration of neighbors that could not be seen.

REFERENCE

[1] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei and S. Savarese, "Social LSTM: Human Trajectory Prediction in Crowded Spaces," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 961-971.

[2] Helbing, Dirk, and Péter Molnár. “Social Force Model for Pedestrian Dynamics.” Physical Review E 51.5 (1995): 4282–4286. Crossref. Web.

[3] M. Everett, Y. F. Chen and J. P. How, "Motion Planning Among Dynamic, Decision-Making Agents with Deep Reinforcement Learning," 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, 2018, pp. 3052-3059.