CS230 Project Report

Trajectory Prediction with Deep Learning

Maneekwan Toyungyernsub
maneekwt@stanford.edu

1 Introduction

There has been numerous research being conducted in the field of autonomous driving. However, technical challenges still exist in the field of control and navigation in both crowded scenes in an urban environment as well as driving on highways. One of the key challenges is to be able to predict trajectories of other traffic agents accurately. These include the motion of pedestrians, bicyclists, and other vehicles including both human-driven and autonomous vehicles. Accurate trajectory prediction will be valuable to the trajectory planning system as well as for long-timescale decision making to ensure safe interaction with other agents on the road.

In this project, we investigate trajectory prediction of different traffic agents with Recurrent Neural Networks (RNN). The input to our algorithm is a sequence of observed trajectories (x,y- coordinates) of various traffic agents. We then use neural networks to output a predicted sequence of future trajectories for a number of timesteps. The challenges will be to explore different movement patterns and predict the trajectories of different traffic agents, including pedestrians, bicyclists, and other vehicles, with accuracy. Traffic agents belonging to different categories typically behave differently, and those that belong to the same category may or may not respond in the same manner under the same circumstances at all times.

2 Related Work

There are many methods to tackle the problem of trajectory prediction. One approach is driver-behavior modeling using deep learning and hidden Markov models have been successfully applied to model driver behavior using a large amount of driving data [1]. Various aspects of driver behavior include lane changing behavior, driver-pedal operation, and car-following behavior. The model can later be used to predict the trajectory of driver's vehicle during a lane change, for example. This approach is specific to vehicle trajectory prediction. Another approach combines a trajectory prediction method based on a constant yaw rate and acceleration (CYRA) motion model and another trajectory prediction method based on maneuver recognition was presented [2]. The combined method results in a better accuracy for both short-term and long-term prediction. However, the model needs the curvature of the road as input and so for real-time implementation, the varying road geometry can add more complexity and longer runtime.

A data-driven approach such as deep learning has been shown to be successful for an end-to-end trajectory prediction task. The inputs to the algorithm encompass x, y- coordinates, vehicle velocity information, and occupancy grid maps where we can include scene-specific information on the state of the surroundings of the vehicles. Likewise, neural networks can be trained to output the predicted sequence of future trajectory and the evolving occupancy grid maps [3, 4]. Both Convolutional Neural network (CNN) and Recurrent Neural Network (RNN) architecture have been used widely in many literatures. RNN is suitable for sequence-to-sequence modeling with time series data, and its Long Short Term Memory (LSTM) variant could retain long-term dependencies and avoids the vanishing and exploding gradient problems. The RNN Encoder-Decoder with Conditional Variational Autoencoder (CVAE) architecture is an extension of the RNN-based approach on trajectory prediction and is capable of modeling multi-modality of trajectories by outputting multimodal probability distributions over possible future actions [4]

3 Dataset

The publicly available dataset from Stanford Trajectory Forecasting Benchmark [5] is used in this project. The dataset consists of many text files of various crowded scenes and contains 2D Cartesian coordinates, frame IDs, and unique object IDs. Drones and surveillance camera videos were used to collect the data at various outdoor spaces throughout the Stanford campus. The data has been manually labeled and processed such that the position coordinates of pedestrians, bikes and vehicles were extracted from the images, and are represented in the world coordinate frame as 2D Cartesian coordinates. The input to our neural networks is a sequence of observed trajectories (x,y- coordinates) for 5 time steps and the model outputs a predicted sequence for the next 5 time steps.

Before training the neural network, all position coordinates in the training set are normalized by its mean and variance, and these same values of mean and variance used to normalized during training are used to normalize the dev and test dataset. The dataset is organized such that each example consists of all valid trajectory data from one unique object ID across multiple time steps.

4 Methods

Network Architecture

Fig.1 illustrates the network architecture proposed for this study. The network is the LSTM Encoder-Decoder architecture which employs two LSTM networks, called the encoder and decoder respectively. The network aims to model the conditional probability of the output sequence given the input trajectory sequence. The encoder processes and provide the summary of the input sequence through the LSTM cell state. We also employ three fully-connected layers and four fully-connected layers in the encoder and decoder sections respectively. Batch normalization is applied after each layer to reduce the covariate shift. The network was trained using Adam optimizer with a learning rate of 0.00005 to minimize the L2 loss between the predicted and ground truth trajectory sequence. The training and test errors are calculated based on the normalized coordinates but the evaluation metrics (to be further discussed in the next section) are calculated based on the unnormalized coordinates in the world frame for better interpretability. Various hyperparameters have been tested and the ones that result in the lowest loss (better performance) are used for actual training. We also train the network developed based on vanilla RNNs model to compare the results with the proposed network architecture.

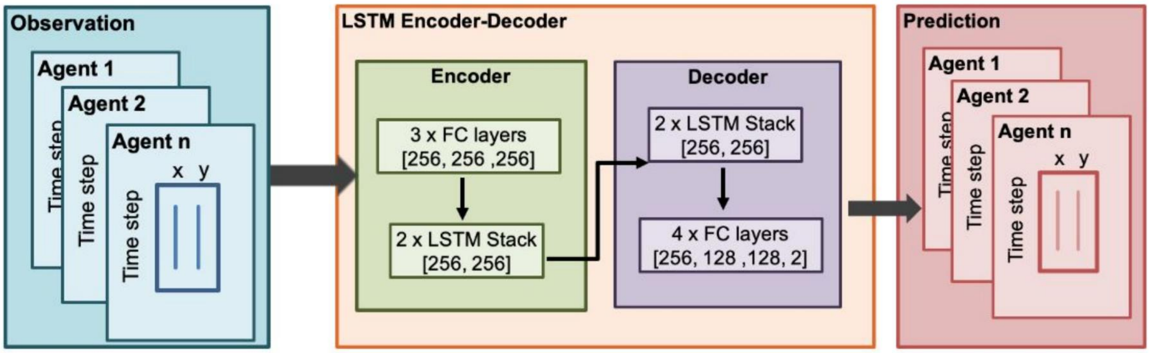


Figure 1: Proposed network architecture

Evaluation metrics

To measure the performance of the models, we can evaluate the mean displacement error as well as the final displacement error. The first metric is taking an average of the Euclidean distances of all predicted positions and true positions during the prediction time. The final displacement error is a measure of the average Euclidean distance between the final predicted positions and the true positions.

5 Results and Discussion

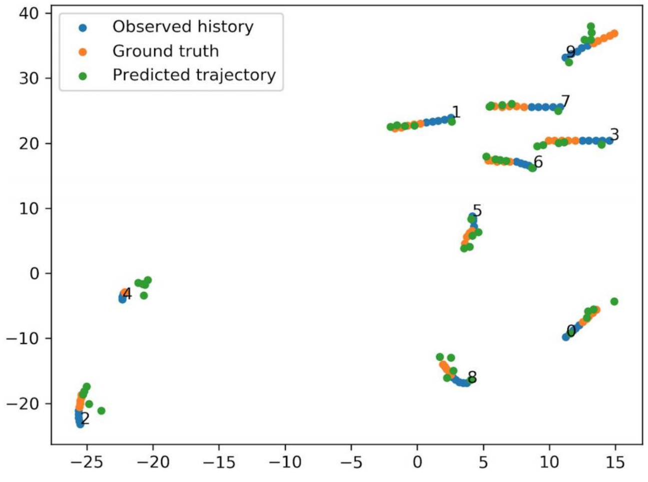
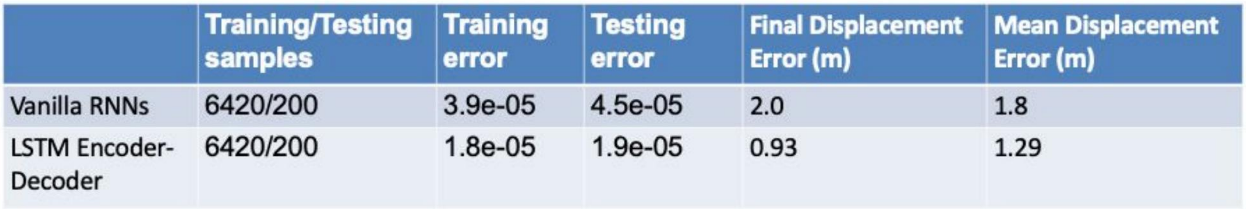


Figure 2. Typical trajectory plots

The results associated with the two models are summarized in Figure 2 and Figure 3. We can see that the LSTM Encoder-Decoder model performs better than the vanilla RNNs model. Training for both models are conducted for 4000 epochs with a mini-batch size of 128 samples. From Figure 2, we can see that the model predict the trend of the motion generally accurately but suffers when there is an uncertainty in the motiont that involves turning and/or moving very slowly or staying stationary (for pedestrains). The challenge is in trying to capture the human-human interaction that influences the their trajectory and hence its prediction. The model may perform better with a larger dataset, which can be for future work where we can use a larger dataset such as NGSIM and KITTI.



	Training/Testing samples	Training error	Testing error	Final Displacement Error (m)	Mean Displacement Error (m)
Vanilla RNNs	6420/200	3.9e-05	4.5e-05	2.0	1.8
LSTM Encoder-Decoder	6420/200	1.8e-05	1.9e-05	0.93	1.29

Figure 3. Training and testing results

Future Work

- Extend the study to larger dataset
- Conduct extensive comparative study with other methods
- Conduct study on incorporating scene-specific information
- Account for multimodality of trajectory prediction

References

[1] C. Miyajima and K. Takeda, “Driver-Behavior Modeling Using On-Road Driving Data: A new application for behavior signal processing”, *IEEE Signal Processing Magazine*, vol. 33, issue 6, pp. 14 - 21, Nov. 2016.

[2] A. Houenou, P. Bonnifait, V. Cherfaoui, and W. Yao, “Vehicle Trajectory Prediction based on Motion Model and Maneuver Recognition”, in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov. 2013.

[3] S. Hoermann, M. Bach, and K. Dietmayer, “Dynamic Occupancy Grid Prediction for Urban Autonomous Driving: A Deep Learning Approach with Fully Automatic Labeling”, in *2018 IEEE International Conference on Robotics and Automation*, May 2018.

[4] E. Schmerling, K. Leung, W. Vollprecht, and M. Pavone, “Multimodal probabilistic model-based planning for human-robot interaction” in *2018 IEEE International Conference on Robotics and Automation*, pp. 1-9, May 2018.

[5] A. Sadeghian, V. Kosaraju, A. Gupta, S. Savarese, and A. Alahi, “Trajnet: Towards a benchmark for human trajectory prediction”, *arXiv preprint*, 2018.