Prediction of Pedestrian Trajectories

Mingchen Li, Yiyang Li, Gendong Zhang

{limc,yiyang7,zgdsh29}@stanford.edu

CS229 Machine Learning, Stanford

Introduction

Prediction on trajectories of pedestrians becomes more essential than ever in the era of the development of autonomous vehicles [1]. We present comparison between performances of different methods on predicting x-y coordinates based on sequential time steps.

Long-Short Term Memory networks (LSTM), Gated Recurrent Units (GRU) and K Nearest Neighbors (KNN) combined with linear regression have been tested. As the results, LSTM and GRU out performed KNN significantly, with the cost around 1e-4 on the first two and 1e-2 on the last one. The evaluation matric is the MSE (Mean Square Error) cost function:

$$L(\hat{p},p)=(\hat{p}-p)^2=\frac{1}{m}\sum_{i=1}^{m}\left\|\Delta p^i\right\|_2^2, where\ \Delta p^i=\begin{pmatrix}x^i\\ y^i\end{pmatrix}-\begin{pmatrix}x\\ y\end{pmatrix}$$

Data

We performed our experiments on two publicly available datasets: ETH [2], and UCY [3]. All the csv files of the dataset have the same format. Table 1 is a glance of the Crowds by Example- University.csv. The four rows from top to bottom are frame, label of pedestrian, y coordinate and x coordinate.

Table 1: A Glance of the Data in Crows by Example-University.csv

Frame	0	10	20	30	40	50	60	70
Pedestrian Label	1	1	1	1	1	1	1	1
Y	0.56076	0.55513	0.5495	0.54387	0.53733	0.52865	0.51997	0.51128
X	0.59722	0.622	0.64677	0.67155	0.6971	0.72447	0.75184	0.77921

Features & Models

The features we used for the models are the sequential steps of people, grouped by labels. However, different filter methods are performed on different models. For LSTM and GRU models, the features are the first number of steps of each person throughout entire dataset. For KNN model, the features only come from the K neighbors of each target person. The number of steps are specified by *num_feature*. Each RNN model contains one layer with 400 nodes. The input and output formats of the models are the same.

For each model, the input is a m**num_feature**2 matrix and the output is m*2*1 matrix, where m is the number labels in the dataset. Figure 1 shows a visualization of the models.

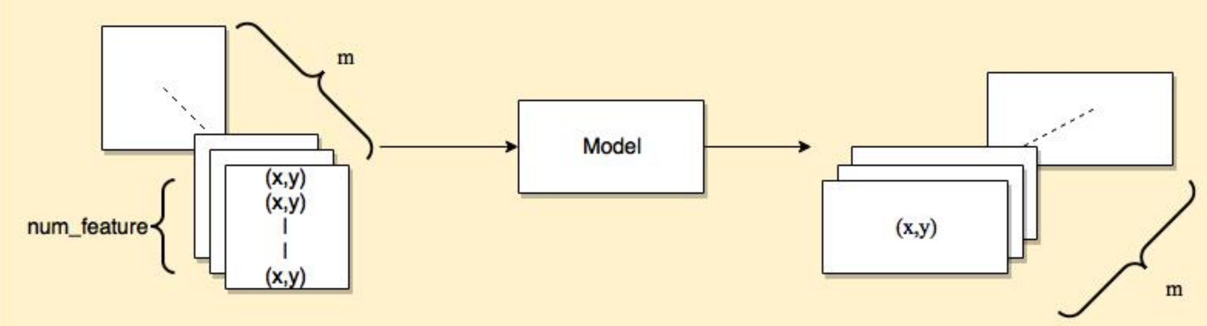

Figure1: Visualization of the models

Result

Table 2 shows the results of different models, where the first two rows of each model represents the correct coordinates, and the following two are the predicted values. It can be seen that LSTM and GRU outperformed KNN by approximately 10 orders of magnitude. In addition, LSTM used 50 epochs to converge, and the cost was around 2e-4, while GRU took 15 epochs to converge, and the cost was approximately 5e-4.

Table 2: Results of different models

LSTM						Average Cost
0.85	0.77	0.74	0.93	0.84	0.77	1.917e-4
0.10	0.16	0.19	0.19	0.21	0.24	
0.84	0.77	0.75	0.92	0.84	0.77	
0.11	0.17	0.19	0.19	0.22	0.24	
GRU						
0.86	0.77	0.73	0.93	0.84	0.76	5.226e-4
0.09	0.15	0.19	0.18	0.20	0.23	
0.84	0.77	0.75	0.92	0.84	0.77	
0.11	0.17	0.19	0.19	0.22	0.24	
KNN						
0.84	0.77	0.75	0.92	0.84	0.77	3.031e-2
0.11	0.17	0.19	0.19	0.22	0.24	
0.58	0.54	0.51	0.64	0.59	0.54	
0.38	0.36	0.35	0.44	0.41	0.39	

Discussion

We performed analysis on the models of RNN and KNN. In particular, we plotted different feature selections versus the cost on the development dataset (Table 3, Figure 2), learning curves and visualization of prediction (Figure 3).

Table 3: Learning rate, batch size vs. cost

num_feature: 5 (loss*1e-3)	Train loss	Test loss	Train loss	Test loss	Train loss	Test loss	Train loss	Test loss	Train loss	Test loss
	0.02		0.005		0.002		0.0005		0.0002	
25	4.47	3.97	2.30	1.96	1.85	1.46	1.65	1.82	2.76	2.88
55	9.02	12.3	3.48	3.43	5.36	5.09	1.86	2.11	1.95	2.29
110	7.47	9.15	4.50	4.49	3.59	3.79	2.61	3.44	3.66	4.00

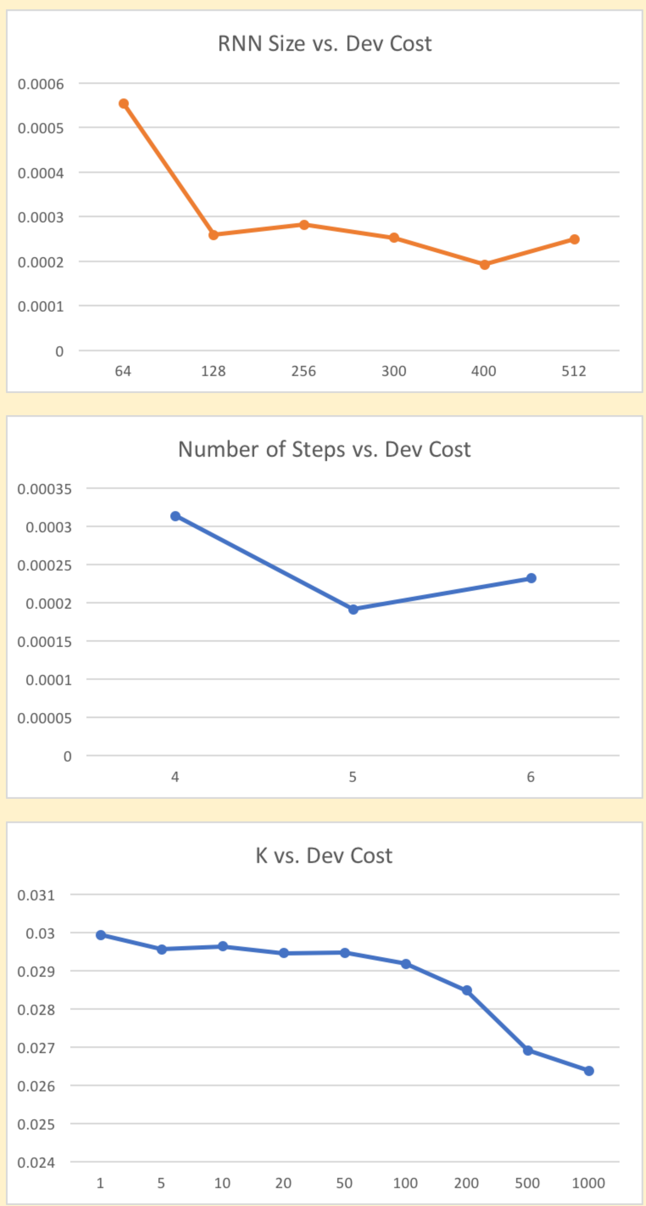

Figure2: RNN size, *num_feature*, K vs. cost

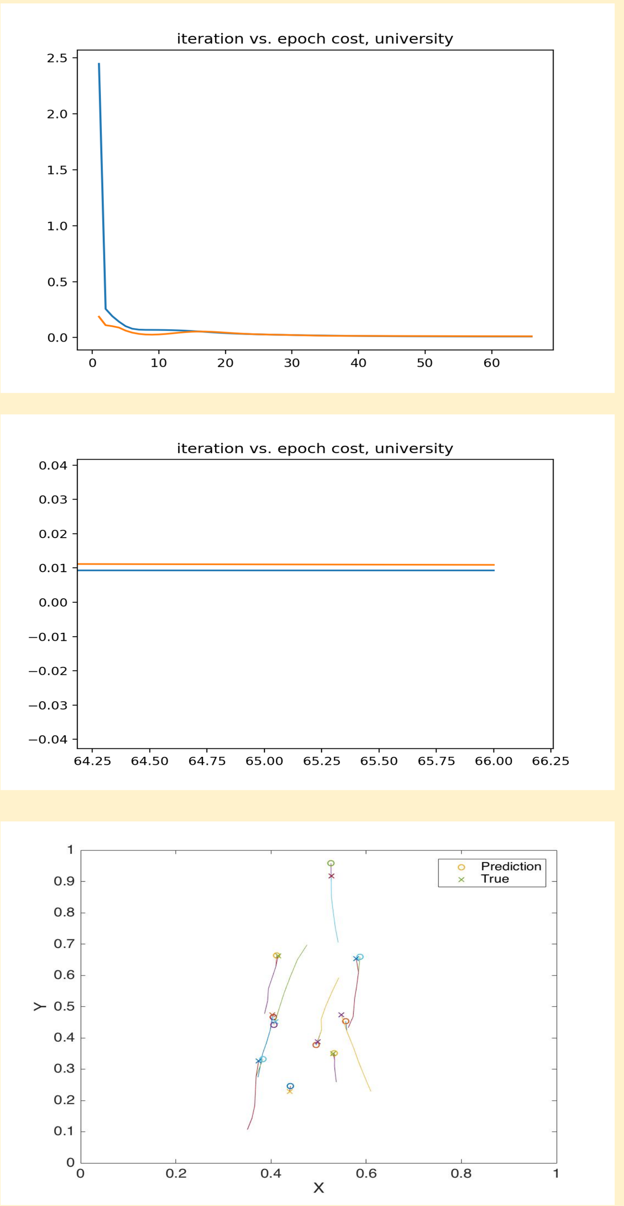

Figure 3: Leaning curve, prediction

Future

- Predict the next few points instead of only one point, implementing sequential-to-sequential prediction.
- Use embedding data processing method to increase the accuracy further.

Reference

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

[2] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In Computer Vision, 2009 IEEE 12th International Conference on, pages 261-268. IEEE, 2009. 2, 5, 6, 7, 8.

[3] A. Lerner, Y. Chrysanthou, and D. Lischinski. Crowds by example. In Computer Graphics Forum, volume 26, pages 655–664. Wiley Online Library, 2007.