UCSD Deep Learning Class Competition

Autonomous vehicle motion forecasting challenge

Team: Chilli Chicken
Sky Li, Michael Chen, Haitong Chen, Yining Gu

Introduction & Summary

Our Team

- Sky Li - sophomore student majoring in Bioinformatics
- Michael Chen - sophomore student majoring in Data Science
- Haitong Chen - junior student majoring in Computer Science
- Yining Gu - junior student majoring in Computer Science

Overview

- Tasks
- Methodology: data processing, training data generating and deep learning model
- Four experiments
- What we learned from the model and the competition
- Future work about the model and the impacts

Task:

City	Train data size	Test data size
Austin	43041	6325
Miami	55029	7971
Pittsburgh	43544	6361
Dearborn	24465	3671
Washington-DC	25744	3829
Palo-alto	11993	1686

Given:

5 seconds of trajectories of vehicles

Goal:

Give reliable predictions of positions of those vehicles 6 seconds into the future

Notice:
Presence complex movement of traffic agents around the AV (cars, cyclists, pedestrians, etc)

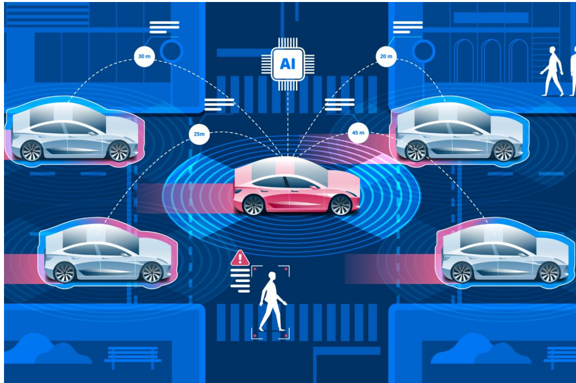
Data given:

Samples from 6 cities that don't have geographic overlap are given. The sample sizes are as outlined in the table above.

Why this is important？

The wide usage of Autonomous Vehicles makes the task of predicting trajectory more and more important.

- Driving safety (collision prediction and prevention)
- Route planning (optimized route)
- Broader impact (will be discussed at the end)



Methodology

Data Processing

Sample wise normalization:

Normalize each sample relative to the first position.

Divide by 100 to reflect changes from the first position

Prevent gradient explosion

Tricks:

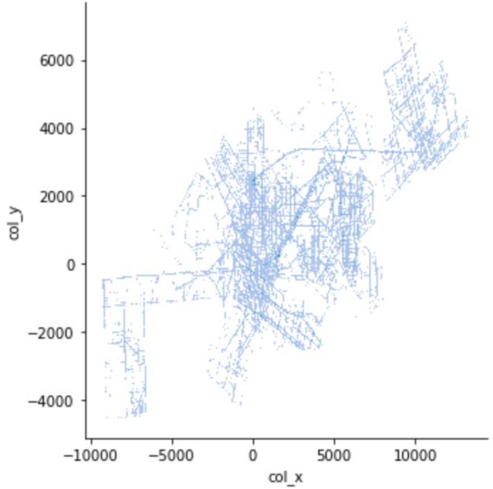
Convert data to from 32 bit to half precision / 16 bit for faster training on Nvidia GPU with tensor core

Generating Training Data

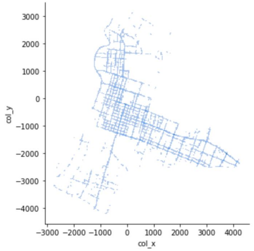
Combine all cities into a larger training set:

(203816, 110, 2): (samples, timestamp, features)

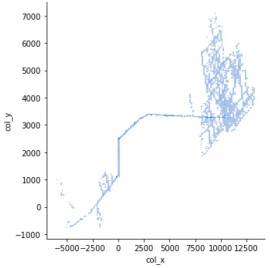
Validation: Training = 1:10



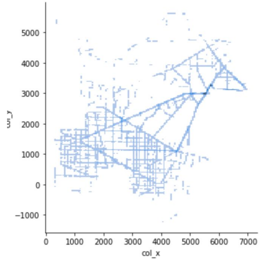
Combined



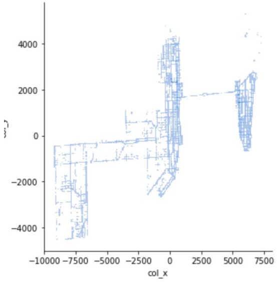
Austin



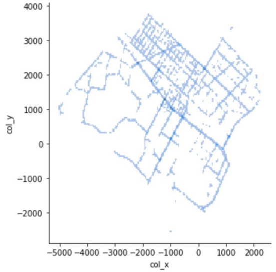
Dearborn



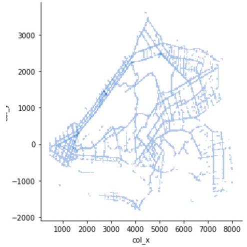
Washington-DC



Miami



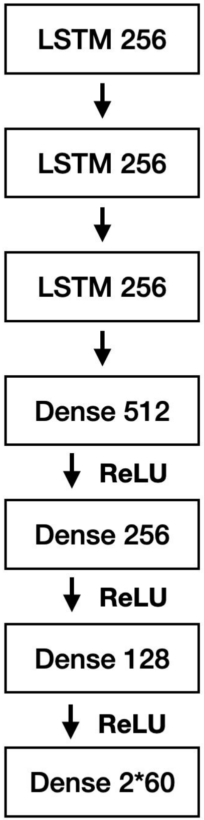
Palo-alto



Pittsburgh

Deep Learning Model

Stacked LSTM + MLP



Optimizer:

Adam

Learning rate:

0.00001

Criterion:

MSE

Batch size:

1

Epochs:

30 epochs

Training time:

2218s / epoch

Experiments

Experiment 1

Temporal Fusion Transformer

Give up half-way

Problems:

Computationally expensive

Too large LR -> diverge
Too small LR -> too slow

Reference:
https://arxiv.org/pdf/1912.09363.pdf

Experiment 2

Vector Autoregression (VAR)

Statistical Model

Problems:

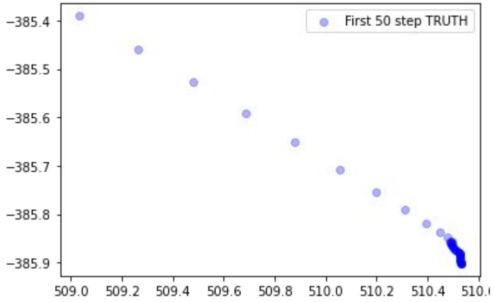
Sparkly large loss data points increases the average loss dramatically

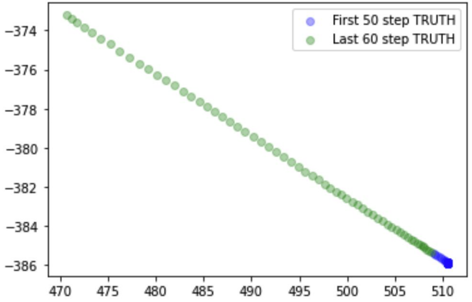
No learning, defy basic physics
(Example: car starting up:)

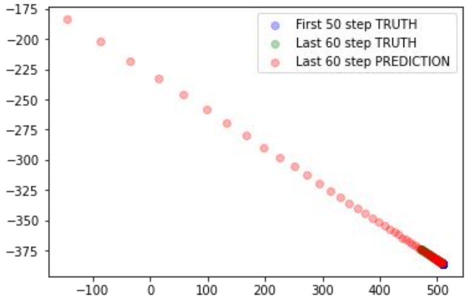
Outcome:

Public score:
48.06760

Private score:
48.29499

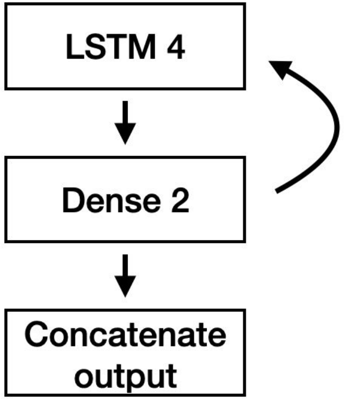






Experiment 3

LSTM, recursive

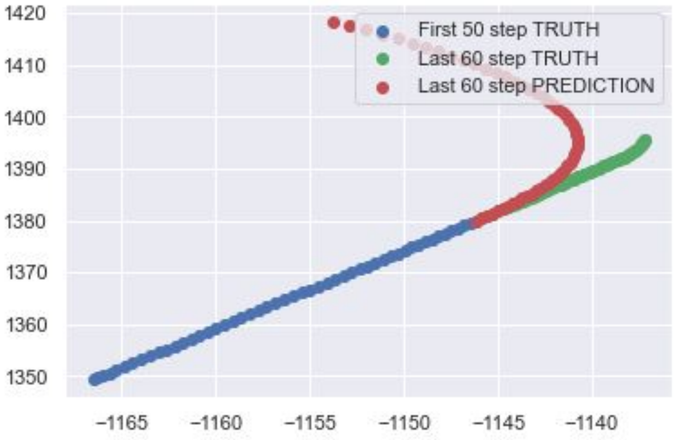


Outcome:

Public score:
371.22272

Private score:
407.31436

Problem:
Error propagation



Experiment 4

LSTM + MLP hyperparameter tuning

Grid search:

LR=[0.001, 0.0001, 0.00001]

LSTM layer=[2,3,4]

MLP layer=[2,3,4]

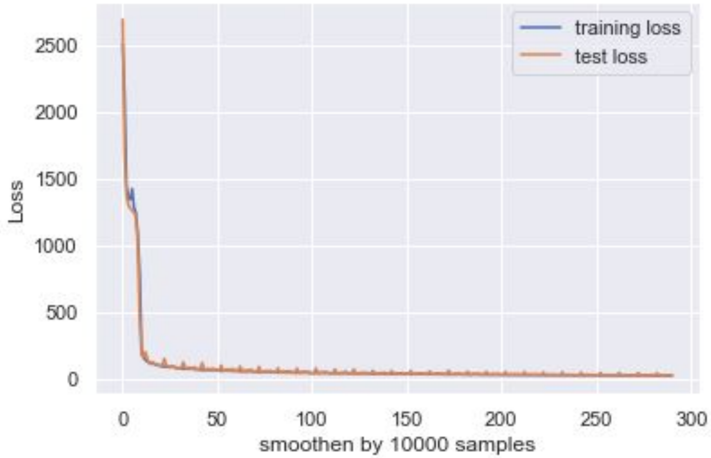
LSTM unit=[32,128,256]

MLP unit=[128,256,512]

Outcome:

Public score:
18.88603

Private score:
19.09141

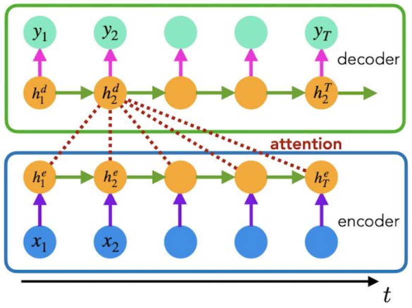


Discussion

What we have learned

The Model:

- Attention based model can potentially bring out better performance during training, but in higher computational cost (time, processing units, etc)
- Different models / different parameters chosen can affect the precision a lot



The Competition:

- Start early! Plan ahead!
- Participate more!
- Use what we've learned in class & internet resources to improve our work.

Future Work

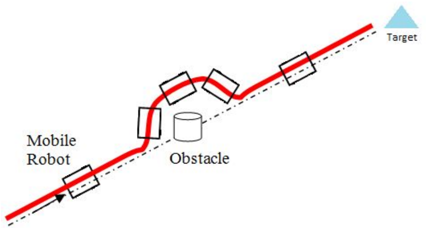
The Model:

- Incorporate feature engineering... but good feature choices!
- Increase the efficiency of the model by shrinking the model size, find better models, etc, so that it has faster responsive predictions which can be better adapted to solve real-world problems.

Impacts:

- Direct impact: better performance of AV in the learning perspective (AI in the computing perspective)
- Broader impact: solve similar forecasting problems to help with more aspects: navigation, obstacle avoidance, etc (IoT integrated)

An example of possible usage in IoT: Using the reliable model, with sensors equipped on human bodies, we can improve the state-of-art obstacle sensing equipment to predict potential obstacles. It can help blind people in their daily lives.



Thank you the instruction team for teaching and helping:

Professor:	Rose Yu
TAs:	Zheng Ding
	Mayank Sharan
	Rui Wang

Questions?