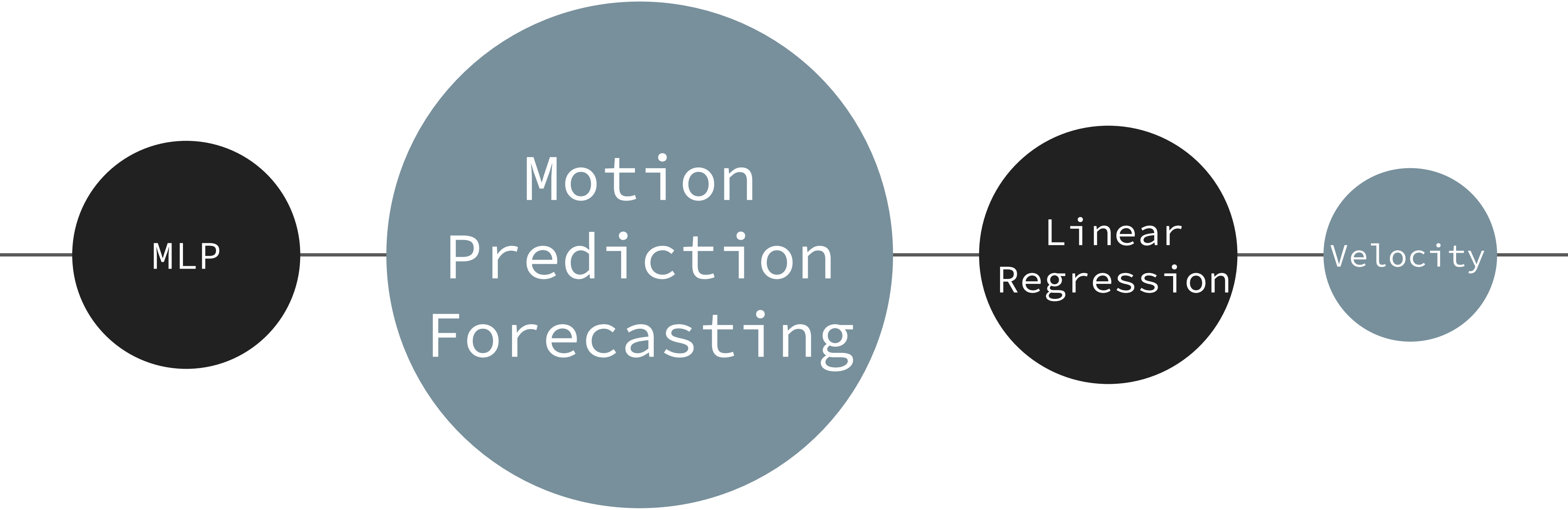
Final Presentation

William Sun

Summary

- Team members: William Sun
- How did you solve the problem: Linear Regression
- Final results:
  - Rank: 12
  - Score: 21.92163
- What have you learned:
  - Implementing and experimenting with multiple ML and DL models
  - Experience with PyTorch, sklearn

Key Words


MLP
Motion Prediction Forecasting
Linear Regression
Velocity

Introduction

Team Introduction

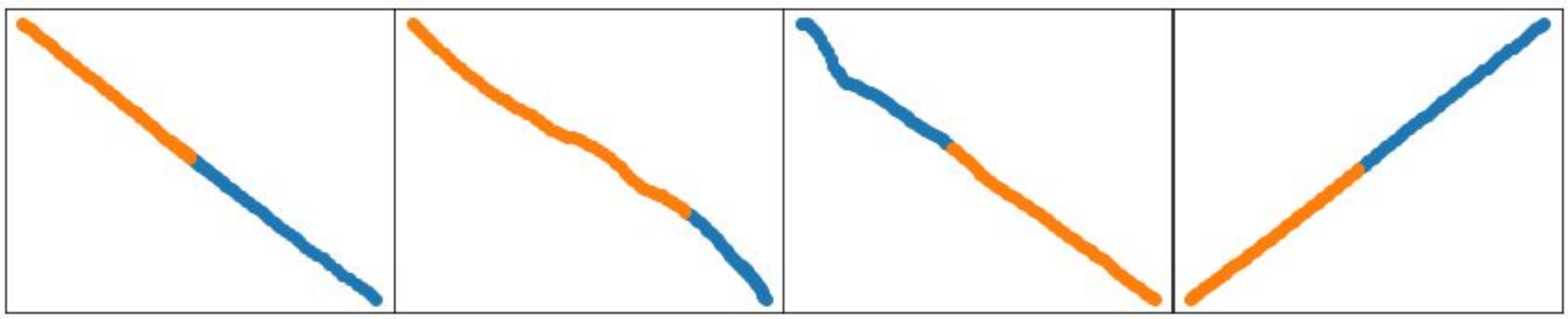
- William Sun:
    - Senior, Computer Engineering
    - Background: CSE 151A, CSE 152A

UC San Diego
Electrical and Computer Engineering
JACOBS SCHOOL OF ENGINEERING

Methodology

Data Processing

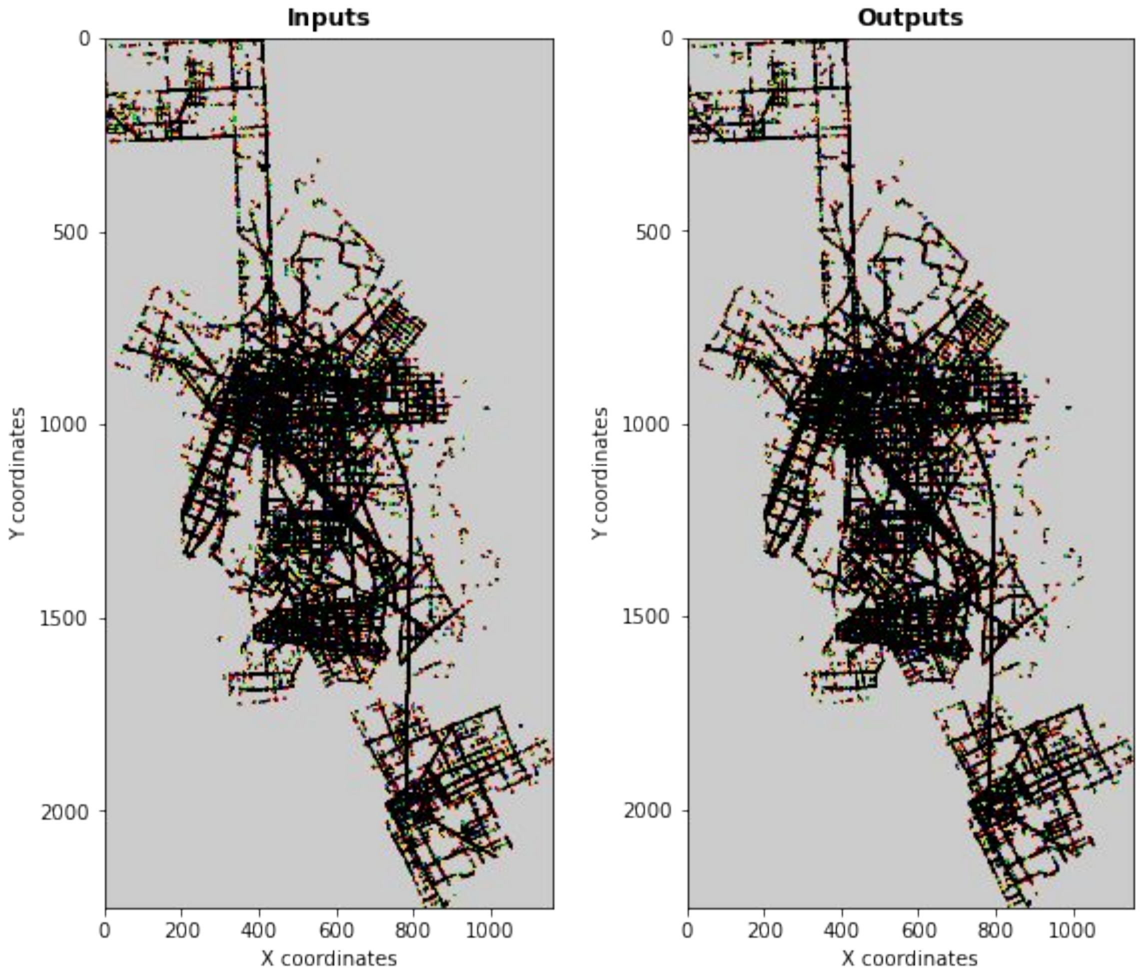
- Summary: the prediction task will consist of taking in 50 positional coordinates which are ordered in time (5 seconds), and output the next 60 predicted positional coordinates in time (6 seconds)
- Initially: 80/20 training/validation split was used
- No normalization was used → performance was satisfactory
- Idea: implement min-max normalization



Visualize sample data batch (Austin #3000)

Data Visualization

- Use imshow() to visualize positions
- Findings:
  - many straight lines
  - densely packed regions

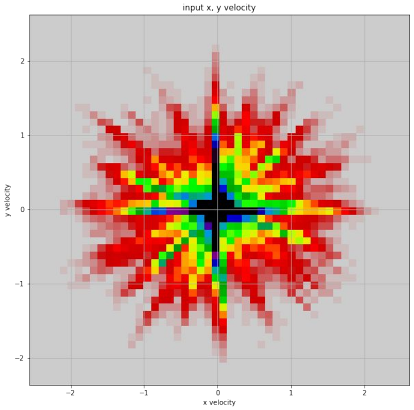


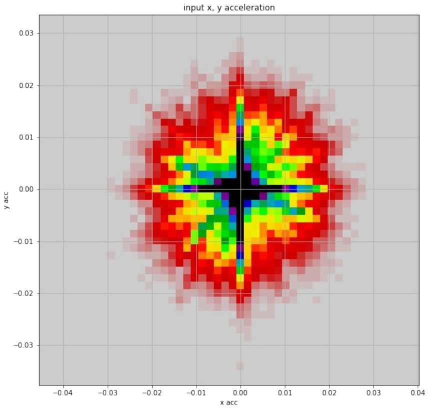
Deep Learning Model

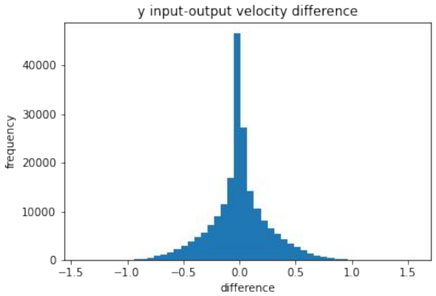
- DL models tested:
  - MLP
  - Encoder/Decoder
  - LSTM
  - Transformer
- Loss function: sum of squares
- Initial hypothesis: more powerful methods would be better (LSTM and transformer)
- Findings: simpler feed forward neural network worked better
  - most data has linear trends, and simple patterns in the data are more geared towards the simpler models
  - complex models not sufficiently tuned during experimentation

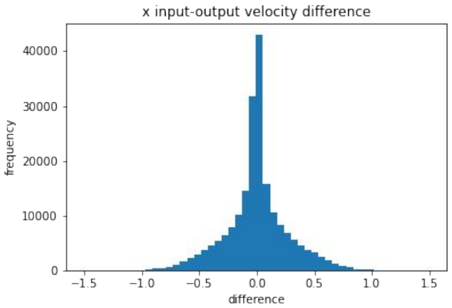
Engineering Tricks

- Idea: take advantage of physics of real-life model → agents must move based on a model of motion
  - Use velocity information
  - Use acceleration information
- Visualize trends of velocity and acceleration in input and output data
- Visualize relationship between input and output data
- Findings:
  - most data points to linear movement (near-zero acceleration
  - input and output data has little difference (constant change)





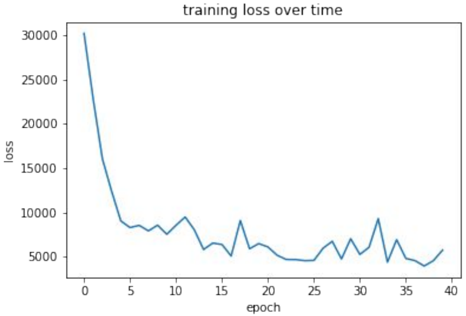


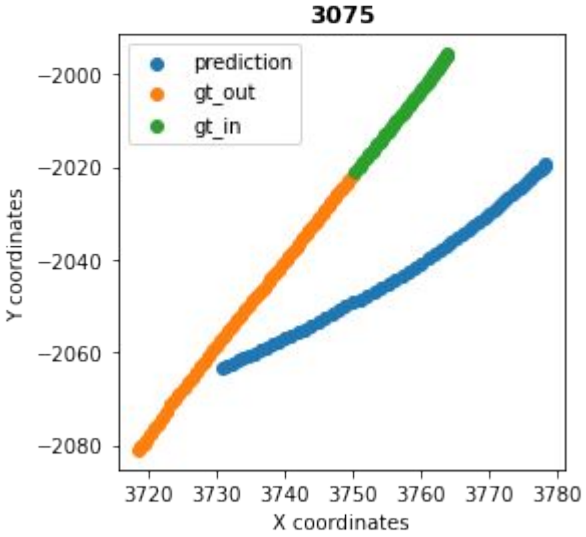


Experiments

Experiment 1

- Encoder/Decoder Model: inspired by discussion 7 architecture
- Parameters:
  - Encoder - 4 hidden layers, Decoder - 4 hidden layers, Batch-size - 4, lr - 1e-3
- Findings: relatively inaccurate (test loss ~1000)



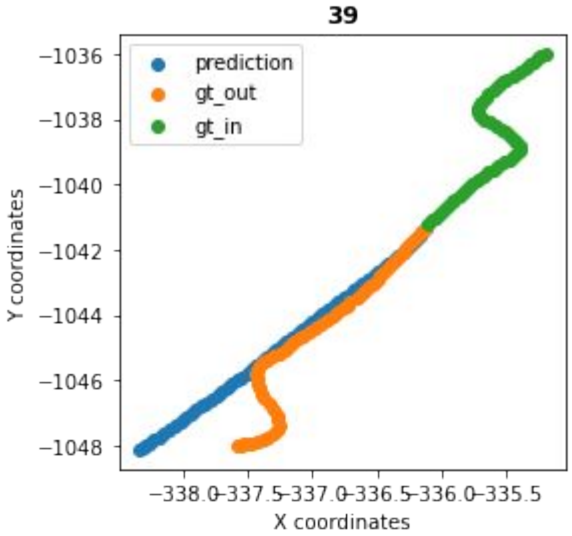


Experiment 2

- Median vs Mean
  - Use same velocity model with acceleration for both tests, just compare median and mean selection
- Model description: calculate the median/mean velocity and acceleration for the 50 input data points, and use this value to interpolate the next 60 future output data points
- Findings:
  - Median performs better (~70 vs ~50 loss)
  - Suggests outliers greatly affect mean model

Experiment 3

- Linear Regression:
  - use 60 separate models total in the linear regression implementation, where each model is prediction the 51-110th timestamps respectively
- Findings: very accurate (test loss ~20)
  - simple is better
  - see image to right, performs weakly against data that has non-linear trends
  - luckily, data visualization shows that most data is linear



Summary of Results

Table 1: Summary of experiment results

Design	Description	Score	Training Time (min per iteration)
(1)	Encoder-Decoder	1733.08	5.2
(2)	Transformer	13579726.63	7.5
(3)	Linear Regression	21.58	0.12
(4)	Average Velocity w/ Acceleration	71.73	N/A
(5)	Median Velocity w/ Acceleration	54.86	N/A

Discussion

What have you learned

- Handling outliers is extremely effective and important for an accurate model based on this data, as shown by the median/mean tests
- Data visualization was most helpful in the early stages of the project
  - Most agents moved in a straight line indicated that a linear model and a simple model would work well
- The biggest bottleneck in this project was setting up the training environment and training itself
- Begin experimentation with the simplest possible models first
  - In this case, simplest performed best!
- Use advice from the reference papers
  - They already provide insights into successful models

Future Work

- With more computation resources and training time:
  - try to tune more complex models to be successful
  - LSTM and transformer models
- Continue tuning hyperparameters for linear regression
- Implement ensemble forest methods for regression
- Apply nearest neighbor approach to test data based on training data
  - May not be effective in real-time, but effective in a competition setting
- Implement more pre/post processing techniques
  - min/max normalization for input data
  - Kalman filter
- Interpolate data to generate more features or collect more data to develop more complex models