CSE 151B Project Milestone Report

Alexander Friend
apfriend@ucsd.edu

Abstract

The abstract paragraph should be indented ½ inch (3 picas) on both the left- and right-hand margins. Use 10 point type, with a vertical spacing (leading) of 11 points. The word **Abstract** must be centered, bold, and in point size 12. Two line spaces precede the abstract. The abstract must be limited to one paragraph.

1 Task Description and Exploratory Analysis

1.1 Problem A [1 points]

Describe in your own words what the task is and why it is important. Define the input and output in mathematical language and formulate your prediction task. Refer to week 1 lectures if you need help.

The task for this Kaggle competition is to predict the positions of 60 individual vehicles 3 seconds into the future, given an initial 2 second observation. This is an important task as autonomous vehicles (AV) are being increasingly rolled out to the public, and are expected to become the future standard of automobile transportation. In order for this future to be realized however, AV's must be able to predict future movement and positions of objects in their visinity with high accuracy in order for AV's to be safer than human drivers.

Our data is split into two sets, a training set and test set, title `new_train` and `new_val_in`, respectively. The training dataset contains the following fields:

- `p_in` - the (x,y) position input in the first two seconds (19 time steps)
- `v_in` - the (x,y) velocity input in the first two seconds (19 time steps)
- `p_out` - the (x,y) position output in the next three seconds (30 time steps)
- `v_out` - the (x,y) velocity output in the next three seconds (30 time steps)
- `track_id` - the track_id for each vehicle in the output sequence (30 time steps)
- `scene_idx` - the id of this scene
- `agent_id` - track id for the agent in this scene
- `car_mask` - boolean index for the real car, we need to align the car numbers
- `lane` - (x,y,z) for centerline nodes in this scene (z position is always 0)
- `lane_norm` - (x,y,z) the direction of each lane node (z position is always 0)

The test set contains the same fields, except it is lacking the `p_out` and `v_out` fields, as these are the fields to be predicted.

Using these datasets, our input will be the the positions (`p_in`), velocties (`v_in`), and ID's of 60 other vehicles (`track_id`) in the scene, as well as information on the position of the lanes. Lane information will include the position of the center of the lanes ((`lane`)) in the scene and the direction of each lane (`lane_norm`). There can be more than one lane in a scene. Using all

Preprint. Under review.

the data in this input set would result with 19 position and velocity values for each vehicle over a period of 2 seconds and $(k \times 3)$ values for each k lane position(s) and direction(s). This gives us $(2 \times (19 \times 2) \times 60) + (2 \times 3k) = 4566k$ dimensions in each input datapoint for k lanes in each scene.

The output will be the ID of each of the 60 other vehicles in the scene (`scene_idx`) with their respective positions (`p_out`). The output will have 10 records/second for the 3 second prediction period for each of the 60 vehicles. This means that the output csv file will have $(10 \times 3 \times 60) + 1 = 3201$, rows including the header, as each row contains the `scene_idx`, and `p_out` of all the vehicles in that time frame. It will therefore have 61 columns, one column for each vehicle position (`p_out`) as well as a column for the corresponding scene id (`scene_idx`).

1.2 Problem B [1 points]

Run the provided Jupyter notebook for loading the data. Provide exploratory analysis on the data and report your findings with texts and figures. Your report should answer the following questions at a minimum:

- **what is the train/test data size, how many dimensions of inputs/outputs**
- **what is the distribution of input positions for all agents (hint: use histogram)**
- **what is the distribution of output positions for all agents (hint: use histogram)**
- **what is the distribution of velocity (magnitude) of all agents and the target agent**

If you include more exploratory analysis beyond the above questions that provides in- sights into the data, you will receive bonus points.

The training data set contains $205,944$ training values, each of dimension $2 \times 60 \times 19 \times 4$. Each training value contains the input and output positions and velocities of each of the 60 cars 1.9 seconds, at a sampling rate of 10Hz.

The validation set contains $3,200$ values, each of dimension $2 \times \times 60 \times 19 \times 4$. Each training value contains the input and output positions and velocities of each of the 60 cars 1.9 seconds, at a sampling rate of 10Hz.

2 Deep Learning Model and Experiment Design

2.1 Problem A [1 Points]

Describe how you set up the training and testing design for deep learning. Answer the following questions:

- **What computational platform/GPU did you use for training/testing?**
- **What is your optimizer? How did you tune your learning rate, learning rate decay, momentum and other parameters?**
- **How did you make multistep (30 step) prediction for each target agent?**
- **How many epoch did you use? What is your batch-size? How long does it take to train your model for one epoch (going through the entire training data set once)?**

Explain why you made these design choices. Was it motivated by your past experience? Or was it due to the limitation from your computational platform? You are welcome to use screenshots or provide code snippets to explain your design.

My model will be a PyTorch *nerual network (make sure to describe actual model)*, running in an Anaconda environment. The platform I am currently working in is my local machine running Ubuntu 20.04 with a 4-core 4-thread Intel i7-7600k CPU running at 4.2 GHz, a GTX 1070 GPU with a max clock speed of 1721 MHz and 8 GB of GDDR5 memory, and 16 GB of 2400 MHz DDR4 memory.

2.2 Problem B [1 Points]

Describe the models you have tried to make predictions. You should always start with simple models (such as Linear Regression) and gradually in- crease the complexity of your model. Include pictures/sketch of your model architecture if that helps. You can also use mathematical equations to explain your prediction logic.

3 Experiment Results and Future Work

3.1 Problem A [1 points]

Play with different designs of your model and experiments and report the following for your best-performing design:

- **Visualize the training loss (RMSE) value over training steps (You should expect to see an exponential decay).**
- **Randomly sample a few training samples after the training has finished. Visualize the ground truth and your predictions**
- **Your current ranking on the leaderboard and your test RMSE.**

Summarize your current experiment results. If you have tried more than one experiment design, compare all of them in a table/figure. Analyze the results and identify the lessons/issues that you have learned so far. Briefly discuss what you plan to do to improve the performance in the following weeks.