FYP21037

Deep Learning for Autonomous Driving

Ho Chung Hin
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Background</td>
<td>2</td>
</tr>
<tr>
<td>Problem Statement &amp; Objective</td>
<td>2</td>
</tr>
<tr>
<td>Literature Review</td>
<td>3</td>
</tr>
<tr>
<td>Methodology</td>
<td>4</td>
</tr>
<tr>
<td>Schedule and Scope</td>
<td>5</td>
</tr>
<tr>
<td>References</td>
<td>6</td>
</tr>
</tbody>
</table>
Background

In recent years, there were several major breakthroughs in the development of Deep Learning technology, which has brought unprecedented success to numerous fields, such as Computer Vision and Natural Language Processing. Deep Learning enables machines to learn complex patterns by adjusting millions of parameters in some powerful neural networks, which aims at minimizing the error value defined by certain mathematical formulas. These technologies have revolutionized the research of different autonomous driving tasks, shifting the ecosystem from traditional rule-based algorithms to data-driven models. Despite rapid development in the field, people are increasingly aware of the risk brought by fully autonomous driving. The latest incident was reported in the Tokyo Olympics, in which the self-driving shuttle e-Palette collided with a visually-impaired athlete, according to euronews [1]. This might be the reason why the authorities and the people conceive that complete autonomous driving would not be realized in the near future, thus, it is believed that learn-based self-driving cars are not yet ready for use in normal traffic condition. However, as analyzed by the United States Department of Transportation [2], driver behavior or error can be accounted for 94% of car crashes. Thus, it can be seen that long-term driving could place much burden to the human drivers, which could lead to fatal errors in different scenarios. On the contrary, the "cognitive functions" of machines would not wane over time, which could make them more suitable drivers than humans. Therefore, it is of great social interest to improve the self-driving capability in order to support fully autonomous driving.

Problem Statement & Objective

Picking up on the incident that happened in the Tokyo Olympics, one key bottleneck of autonomous driving can be observed - trajectory prediction. According to a report from Forbes [3], the e-Palette stopped as it detected someone at the roadside of a T-junction. However, the overseer inside the shuttle restarted driving, thinking that the athlete would stop as the shuttle was making a turn - unfortunately the visually-impaired athlete did not stop walking as he was not aware of the shuttle. This shows that trajectory prediction is a difficult task even for human drivers.

Trajectory prediction is difficult as there are two major challenges, namely to resolve uncertainties in decision and to capture the complex social interaction on roads. In a paper published by Gilles et. al. [4], it is summarized that there are two types of uncertainties underlying the prediction task, namely Aleatoric uncertainty and Epistemic uncertainty. The former describes the natural randomness observed in variation of physical quantities (e.g. velocity, acceleration) - which can be caused by different types of vehicle, while the latter describes the information that is unknown to the driver e.g. the condition or high-level goal of other road users. On the other hand, the decisions of road users are also affected by their own
interactions. Such interaction is most obvious when the drivers need to take proactive action, e.g. when they are changing lanes or overtaking other traffic participants. However, it is uneasy to capture this relation precisely and efficiently. Therefore, this project aims at designing and implementing an efficient neural network architecture, so as to enhance the ability of machines in handling prediction uncertainties and parsing interaction-heavy traffic scenarios. The model should be evaluated against existing solutions to verify if it produces predictions with lower error rate within a shorter period of time.

By improving the trajectory prediction capability, it is expected that machines can plan safer interactions proactively when it is anticipated that certain accidents are highly probable to follow if no actions are taken. This way, machines are prevented from responding in a reactive manner, which should reduce the number of unsafe behaviours, e.g. hard braking.

**Literature Review**

In a recent review of different trajectory prediction techniques [5], it can be seen that interaction-aware models are replacing numerous types of older models. First, physics-based models lay their predictions on observed physical quantities (e.g. acceleration, weight), then predict the yaw rate or position of an agent. This type of model is sensitive to the data noise of the sensors and are typically only suitable for short-term predictions (e.g. under one second). Second, stochastic-based models lay their predictions on probabilistic theory, and assume that the action of each agent is independent of the others. This type of model is not able to model all the combinatorial possibilities and can make false conditional predictions when there are mistakes in previous predictions. Third, maneuver-based models lay their predictions on patterns mined from training data, then use some multimodal estimation algorithm to make predictions. These trajectory prototypes would not consider the social interaction at the time of decision, which missed the essential context information that human drivers would consider.

The most common structure for these interaction-aware models is an encoder-decoder network. The encoder is typically constructed by stacked layers of Convolutional Neural Network (CNN) or Recurrent Neural Network (RNN), specifically Long Short-Term Memory (LSTM) architecture for RNN. The choice of encoder component is normally coupled with the choice of input data. For CNN, the model typically accepts a vector of image appended by the physical quantities of agents, which the model extracts distinctive features e.g. steering angle (i.e. the wheel turning angle) for prediction. Due to the nature of CNN, they are usually much faster than a RNN model. For RNN, the model accepts sequence data from different agents and explicitly handles the multimodal data. Both RNN and CNN are able to capture social interaction to some extent. For instance, one implementation of the CNN encoder tries to stack convolutional layers on top of each other so that they learn the temporal dependencies of the scene, another implementation of the RNN encoder tries to add pooling layers to the hidden states of multiple neighbouring
LSTMs to combine them together. As for the decoder, it is typically built by Conditional Variational Autoencoder (CVAE), which generates the prediction conditioning on the encoding vector (e.g. the hidden representation from the encoder).

There are some novel models that are built by making modifications on top of these previous models, such as using a Generative Adversarial Network (GAN) to replace the CVAE decoder, using attention mechanism to parse the dependencies hidden in the context, or using Graph Neural Network (GNN) to replace CNN/RNN as the encoder. It is planned that these approaches would also be examined in subsequent reviews.

**Methodology**

The dataset that would be used in this project is called *nuScenes* [6]. *nuScenes* contains about 600GB of data, varying from camera images, map data to RADAR and LIDAR data. The data is annotated at a rate of 2Hz, with one of the twenty-three predefined classes and the 3D bounding boxes of the objects. Moreover, attributes such as visibility, activity, pose, size are also included in the annotations. *nuScenes* is chosen as the dataset for a few reasons. First, it provides traffic data retrieved from both Singapore and the United States. These two places are demographically different in terms of the driving habits or races of the road users, and geographically different in terms of their location, traffic density, and weather. These variations are considered to be essential in order for the model to better generalize for different regions which may have different road markings, lane structures, etc. Second, the dataset is more developer-friendly. A development kit is provided by the same team for convenient retrieval of the data, along with dedicated interfaces for the prediction task and for using map data. This is particularly suitable for this project which aims at developing a model for trajectory prediction. Third, the dataset is actively maintained and extensions are continuously released for training, which indicates that there would be good support in the near future. Fourth, *nuScenes* hosts their own competitions for multiple autonomous driving tasks, such as 3D object detection, tracking and LIDAR segmentation. Hence, numerous performance metrics have already been specified by the team, which makes it convenient for evaluating the performance of the project.

As for the architecture design, it is anticipated that it would be split into four components. First, performing scene preprocessing for the data retrieved, this includes filtering data noises that are irrelevant to the prediction and orientation normalization, etc. These would essentially increase data quality and allow the model to learn better. Second, a history encoder, which tries to capture the time-series information recorded before the moment of prediction. This is essential as the behaviour of the agents would, to some extent, be physically constrained by their own actions recorded in previous time-steps. For instance, the turning angle of a vehicle would determine whether it is able to make the turn at certain intersections. Third, parsing agents interaction so that the social context of the traffic scenes can be encoded into the network and
be learnt in the training stage. As illustrated earlier in the discussion of the car incident that happened in the Tokyo Olympics, it is important to know how the actions of the other agents would affect the observing agent before making decisions about its routes. Forth, a trajectory forecaster that tries to predict the probability distribution of different possible motions, which should also handle the aleatoric uncertainty and epistemic uncertainty mentioned earlier.

As mentioned in the discussion of the nuScenes dataset, several evaluation metrics has been proposed, specifically for the prediction task. The first one is called Minimum Average Displacement Error over k (minADE_k), which is an average value of the pointwise euclidean distances between the predicted trajectories of the model and the ground truth value provided by the dataset, over the k most likely predictions. The second one is called Minimum Final Displacement Error over k (minFDE_k), which is an average value over all agents using the calculation of minimum final displacement error (FDE) over the k most likely predictions. Here, FDE is defined as the euclidean distance between the final points of the prediction and the ground truth value. The third one is the Miss Rate At two meters over k (MissRate_2_k), which is the proportion of misses over all agents in its k most likely predictions. Here, a prediction is classified as "miss" if the maximum pointwise euclidean distance between the predicted value and the ground truth value is greater than two meters. Note that nuScenes allows the model to consider a maximum of two seconds of history data, which would be used to predict long-term (i.e. six seconds) trajectories (i.e. sequence of x- and y- coordinates with respect to the global frame) sampled at 2Hz with at most twenty-five modes (i.e. possible routes) and their probabilities. Here, examples of road agents are pedestrians, cyclists, buses, cars, etc.

**Schedule and Scope**

| Sep 2021       | - Set up the nuScenes dataset and related libraries  
|                | - Meeting with supervisor for ideation and inception  
|                | - Brief literature review for understanding the current development of trajectory prediction |
| Oct 2021       | - Complete compiling the project plan and setting up the project web page  
|                | - Further literature review, particularly on methods to capture social interaction and comparison of different proposed models |
| Nov 2021       | - Further literature review, explore how to integrate novel deep learning components into the existing network  
<p>|                | - Design the network architecture |</p>
<table>
<thead>
<tr>
<th>Month</th>
<th>Activities</th>
</tr>
</thead>
</table>
| Dec 2021  | - Implement the drafted network design  
- Evaluate against baseline models |
| Jan 2022  | - Prepare and deliver the first presentation  
- Complete compiling the interim report  
- Deliver the preliminary implementation |
| Feb 2022  | - Either seek to improve the model or explore route planning  
- Literature review for the chosen task |
| Mar 2022  | - Implement the new design or route planning  
- Evaluate the new implementation or the route planning model |
| Apr 2022  | - Complete compiling the final report  
- Deliver the finalized tested implementation  
- Prepare and deliver the final presentation |
| May 2022  | - Update the finalized project web page  
- Project exhibition |

- **Included**
  - Literature review
  - Deliverables
    - Designing, implementing & testing the architecture
    - Evaluation against existing methods
  - If possible: extension to route planning
- **Excluded:**
  - Business application
  - Integration into existing driving system

**References**


