Motion Prediction for Autonomous driving

Department Of Electrical engineering
IIT (BHU) Varanasi

Presented by

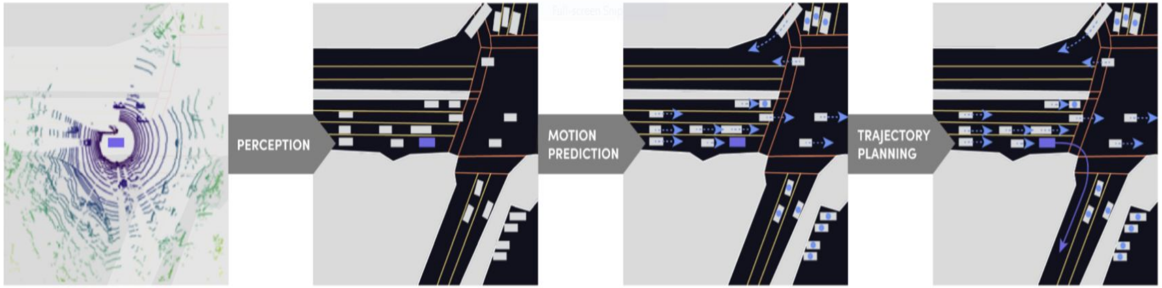
Aloori Raj Aryan 18085005
Bodaballa Vamsi Krishna 18085018

Under the supervision of

Dr. Avirup Maulik
Asst. prof IIT (BHU)
Department of Electrical Engineering

Introduction

Autonomous vehicles (AVs) are expected to dramatically redefine the future of transportation. However, there are still significant engineering challenges like building models that reliably predict the movement of traffic agents around the AV. Motion prediction can be done based on experience and recently observed series of past events, and entails reasoning about probable outcomes with these past observations. Aspects that influence driving behavior comprise many factors, such as general driving physics, infrastructure geometry, traffic rules, weather and so on.



Sensor input and maps | Detected traffic agents | Predicted agent motion | Path taken by autonomous vehicle

Motion prediction

- Accurately predicting the future trajectory of a vehicle not only can reduce or eliminate the collision risk when autonomous vehicles perform complex driving maneuvers, such as merge, lane change, and overtaking, but also can improve the driving efficiency and comfort of autonomous vehicles.
- The vehicle's driving trajectory is not only constrained by prior knowledge, such as that about the road structure, traffic signs, and traffic rules, but also by uncertain posterior knowledge, including subjective driving intentions of the driver.
- To obtain a reliable perception of the environment, the ego vehicle relies on a number of sensors, generally consisting of laser scanners (LIDAR), 3D cameras, regular cameras, radars, and ultrasonic sensors.
- Deep learning-based approaches do not require the designer to define the relations between features and driving behavior. Instead these models try to find the relations in the data that can explain the driving behavior, by looking at example data but requires large amount of data and the availability of large-scale datasets has been a large contributor for the progress.

Lyft Motion Prediction Dataset

The largest self-driving dataset for motion prediction to date, containing over 1,000 hours of data was collected by a fleet of 20 autonomous vehicles along a fixed route in Palo Alto, California, over a four-month period. Here we outline the details of the released dataset, including the process that was used to construct it. An overview of different dataset statistics can be found in Table 1.

The dataset has three components:

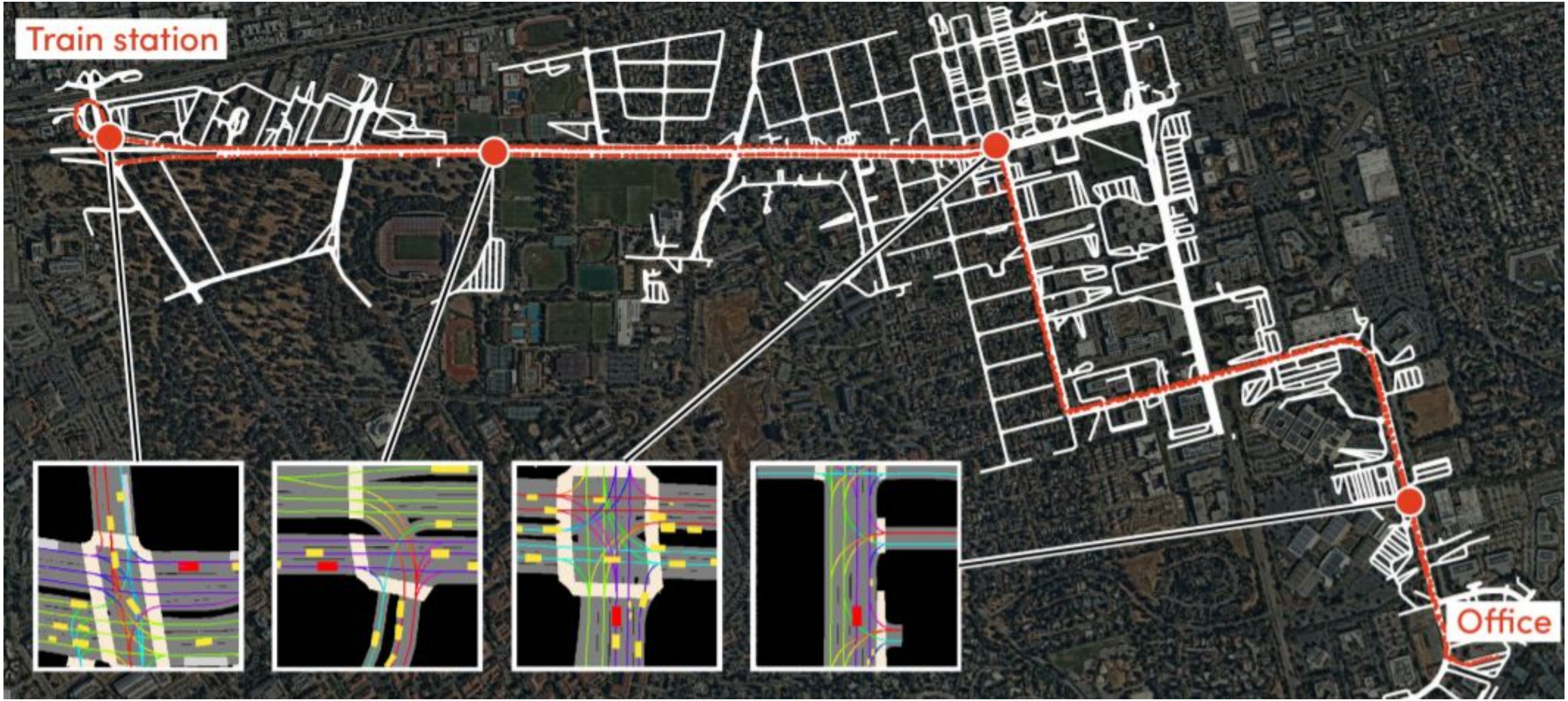
1. 170,000 scenes, each 25 seconds long, capturing the movement of the self-driving vehicle, traffic participants around it and traffic lights state.

2. A high-definition semantic map capturing the road rules, lane geometry and other traffic elements.

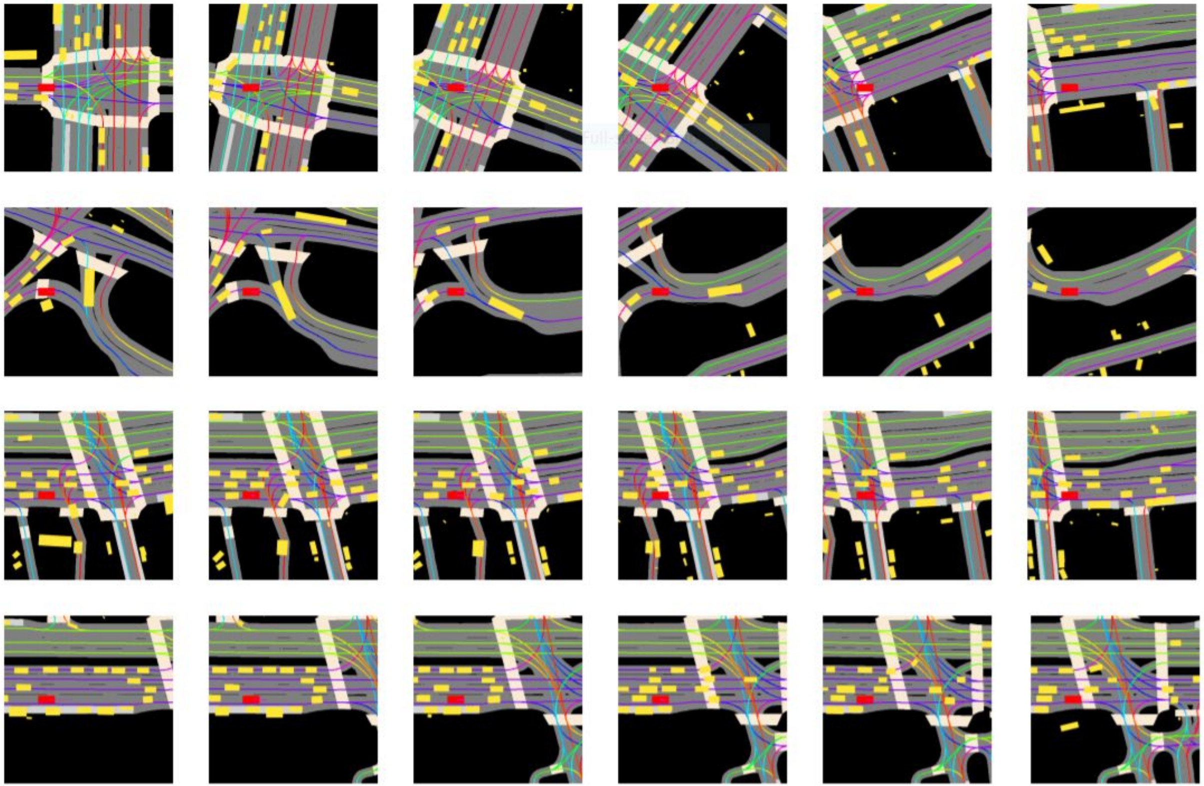
3. A high-resolution aerial picture of the area that can be used to further aid in prediction.

Statistic	Value
# self driving vehicles used	20
Total data set size	1,118 hours / 26,344 km / 162k scenes
Training set size	928 hours / 21,849 km / 134k scenes
Validation set size	78 hours / 1,840 km / 11k scenes
Test set size	112 hours / 2,656 km / 16k scenes
Scene length	25 seconds
Total # of traffic participant observations	3,187,838,149
Average # of detections per frame	79
Labels	Car: 92.47% / Pedestrian: 5.91% / Cyclist: 1.62%
Semantic map	15,242 annotations / 8,505 lane segments
Aerial map	74 km^2 at 6 cm per pixel

Table 1: Statistics of the dataset



High definition (HD) Semantic map of the area



Examples from the scenes in the dataset, projected semantic map. The self-driving vehicle is shown in red, other traffic participants in yellow, and lane colours denote driving direction.

Model

The problem of training very deep networks has been relieved with the introduction of these Residual blocks and the ResNet model is made up of these blocks. The problem of training very deep networks has been relieved with the introduction of these Residual blocks and the ResNet model is made up of these blocks. In the below figure, the very first thing we can notice is that there is a direct connection that skips some layers of the model. This connection is called ’skip connection’ and is the heart of residual blocks. The output is not the same due to this skip connection. Without the skip connection, input ‘X gets multiplied by the weights of the layer followed by adding a bias term.Then comes the activation function, f() and we get the output as H(x).

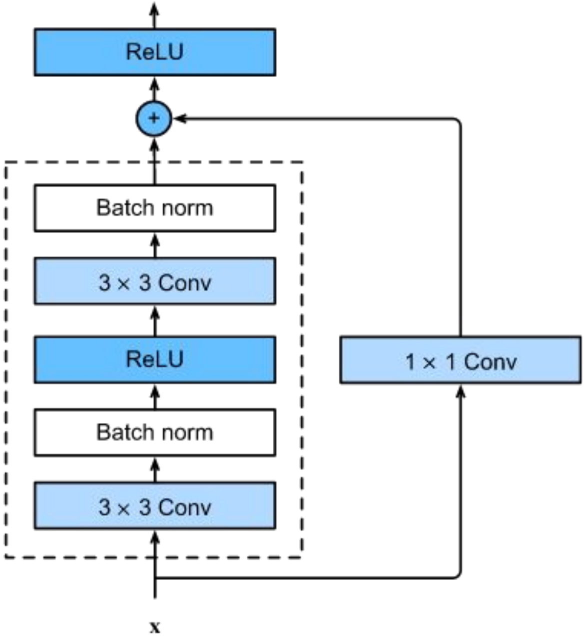
$$H(x)=f(wx + b) \text{ or } H(x)=f(x)$$

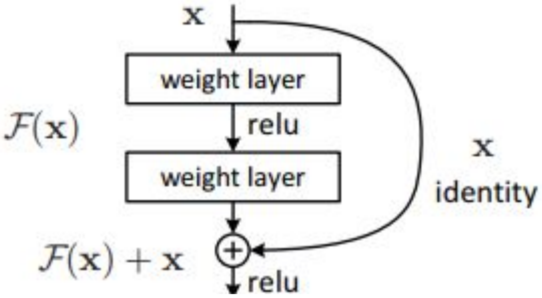
Now with the introduction of a new skip connection technique, the output is H(x) is changed to

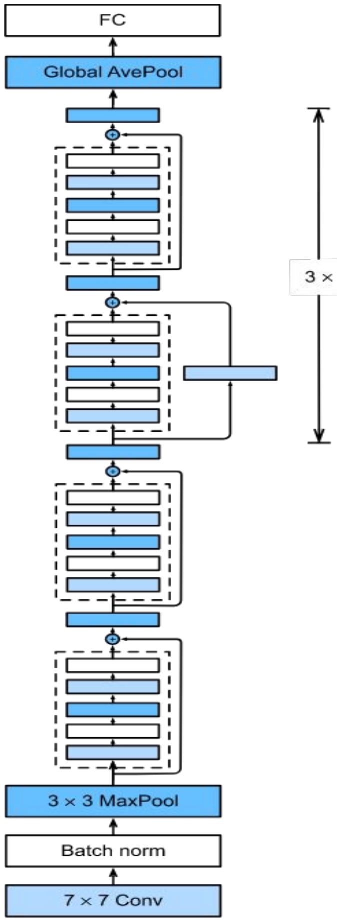
$$H(x)=f(x)+x$$

1×1 convolutional layers are added to the input to match the dimensions. In such a case, the output is:

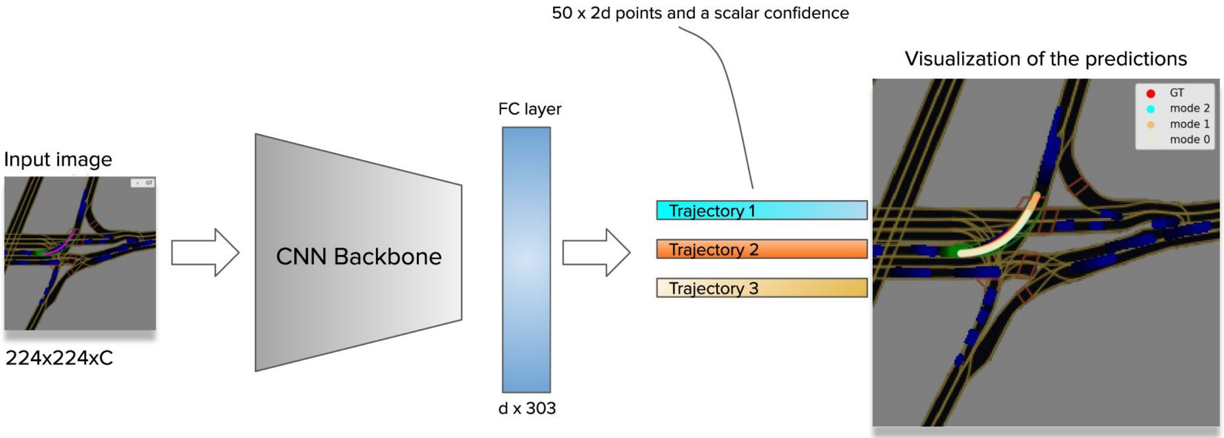
$$H(x)=f(x)+w1.x$$


ReLU
+
Batch norm
3 × 3 Conv
ReLU
Batch norm
3 × 3 Conv
1 × 1 Conv
x


x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu


FC
Global AvePool
3 ×
3 × 3 MaxPool
Batch norm
7 × 7 Conv

- The goal is to predict three possible trajectories of the other traffic agents. We could use the pretrained Resnet 18(or Resnet 32, Resnet 64 etc), a CNN network, and create a multi modal one generating multiple hypotheses(3 in this case) - further described with a confidence vector.
- Due to high amount of multi-modality and ambiguity in traffic scenes, the used evaluation metric to score this model's final predictions is tailored to account for multiple predictions



Loss function and Evaluation metrics

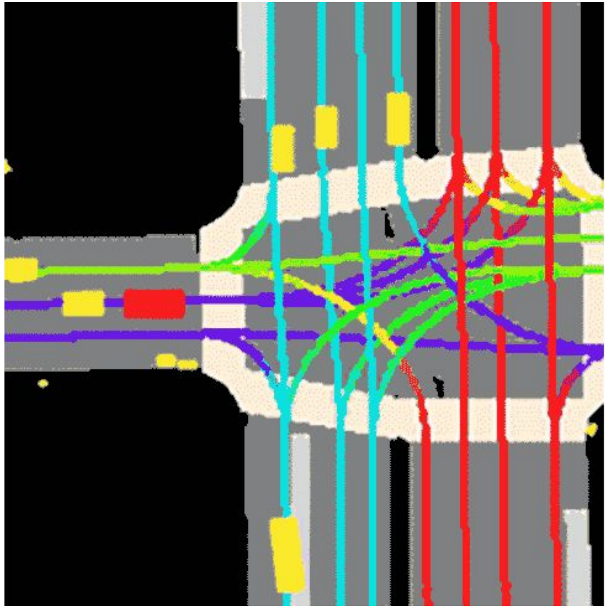
Negative Log Likelihood (NLL): For each data sample in a multi-class classification task, NLL is calculated as:

$$NLL = \neg\sum^{M}_{c=1} y_c \log(\hat{y}_c)$$

Where y_c is a binary indicator of the correctness of predicting the data sample in class *c*, $\hat{y}c$ is the predicted probability of the data sample belonging to class *c*, and *M* is the number of classes.

Computation Time: The trajectory prediction models are usually more complex compared to intention prediction models. Therefore, they can take more computation time which might make them impractical for on-board implementation in autonomous vehicles. Thus, it is crucial to report and compare computation time in trajectory prediction models.

Average Prediction Time: This metric is used in intention prediction approaches such as lane change prediction, where the approach is applied on a sliding window of the input data series to predict the occurrence of a positive class(e.g., lane change). The metric is obtained by taking the average of the time of the first correct positive class prediction for all samples, considering the time of lane change occurrence as the origin. We considered the time when a consistent correct lane change prediction starts to increase the robustness of the metric.



Example of the agent and its history, the agent is AV itself, but the dataset contains similar recordings of other cars as well. Colored lines depict the road lanes

Results and Conclusion

- The whole training process of the train data and prediction on the test data took 7309.7 seconds (or 2.03 hours) with GPU accelerator.
- The final model has a score of 23.515 on the partial dataset (the lower the score the better the model). The score is a combination of the evaluation metrics used.
- The output is CSV file of shape (71122, 305) ie. 71122 rows and 305 columns.
- The output file contains the predicted coordinates of the three possible trajectories along with the associated confidences calculated for each trajectory.
- The output file also contains the track id and the timestamp associated.
- We could improve our model with Hyperparameter tuning using Grid search.
- The model could be improved with ensembling and including LSTM units (RNNs) might significantly improve the predictions. A CNN + RNN model might improve the predictions.
- As we predicted the possible trajectories of the traffic agents, the AV has an understanding of the environment it has to manuever accordingly. This is called Motion planning.

Thank You