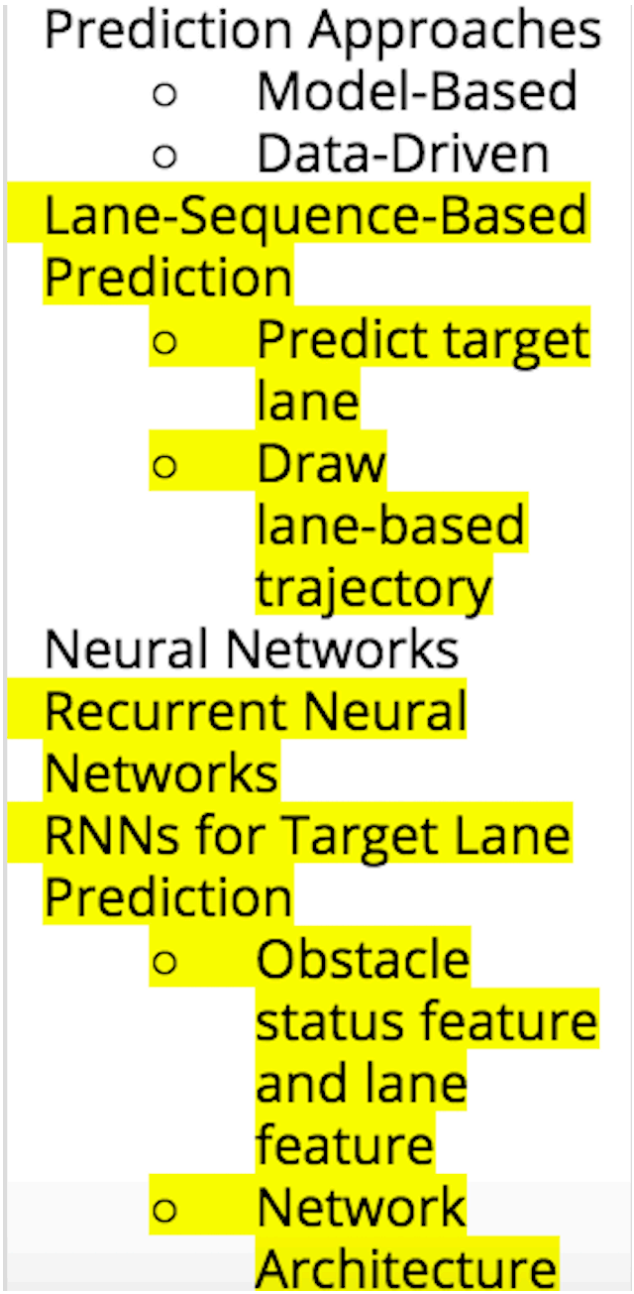
Prediction Module

Assignment



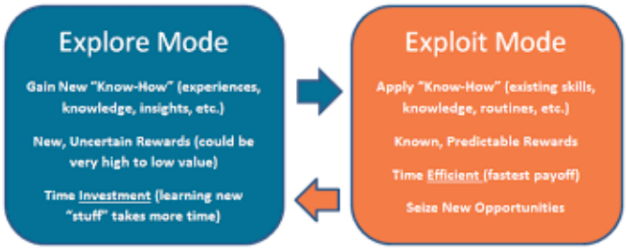
Prediction Module

Lane-Sequence-Based Prediction

Prediction is a common component of IQ tests such tests often have question in the form of what comes next 1 1 2 3 5 8 13 and then the test has to fill in the blanks.

1 1 2 3 5 8 13 [ ] [ ]

In order to predict we shall create an agent in a way could exploit learned data to explore the future.



Predicting other traffic participants trajectories is a key task for an autonomous vehicle, in order to avoid collisions on its planned trajectory.

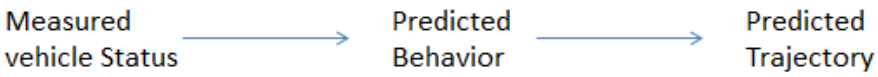
The prediction module receives the obstacles information e.g. positions, headings, velocities, accelerations, and generates the predicted trajectories with probabilities for the obstacles.

Measured vehicle Status → Predicted Trajectory

Moving behavior of road objects is dependent on the the type of object. A vehicle's behavior might be *keep lane* or *change lane* on highway, *make a turn* in urban environment. While cyclists or pedestrians have totally different possible behaviors. so we have to add one more

Prediction Module

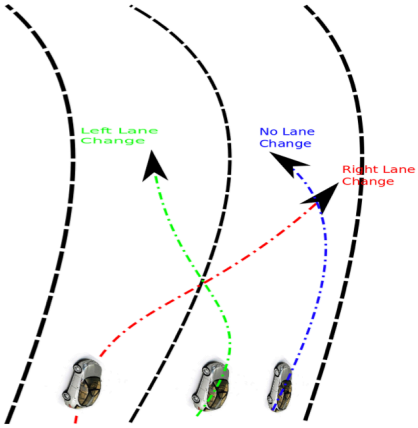
step to make our model even more robust: first predict the behavior and then computes the actual trajectory.



We can classify the behavior by enumerating all possible behaviors but not really. For example, there could be just one lane, or multiple left and right-turn lanes, or even free moving obstacles block all possible lanes. So, we can not choose a proper categories of behaviors based on distinctive maps as a lot of “labels” will lead to an overwhelmingly complicated and not scalable situation.

1. The first behavior
2. The second behavior
3. The third behavior
4. The forth behavior
5. The fifth etc...

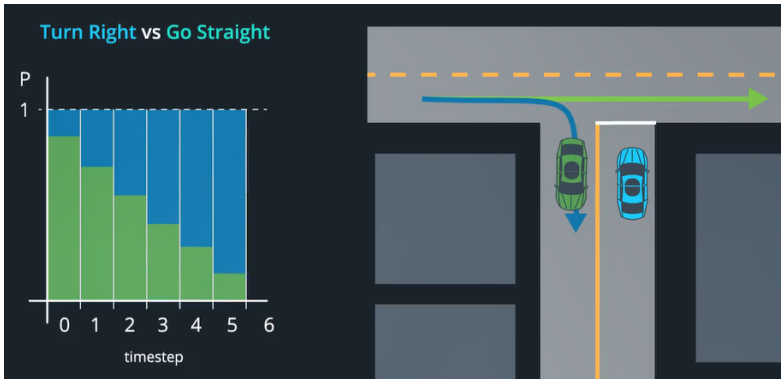
Lane-Sequence is one of the above mentioned scenarios in which vehicle objects move along lanes in a multiply lane landscape thus we impose an assumption that vehicle follows certain logical or structural sequence of lanes upon which behavior is easy to predict (limited maneuver).



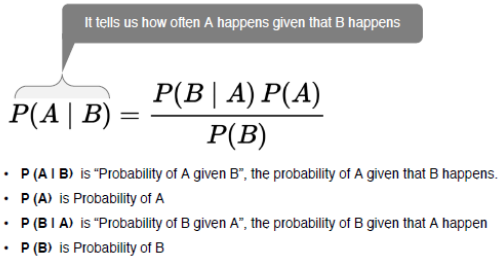
Prediction Module

Predict Target Lane

While detecting other vehicles and localizing them to the map, we focus on developing a framework that uses proper features to predict near future target lane probabilities. i.e. the probability of the predicted maneuver (turn right of go straight) updates according to the observed latest vehicle positions, velocities, etc.



the probability can be calculated by Bayes rule:



if A is defined as maneuver and B is defined as sensed vehicle state.
then P(A|B) will stands for:

the probability of maneuver given the vehicle position and velocity(observation).

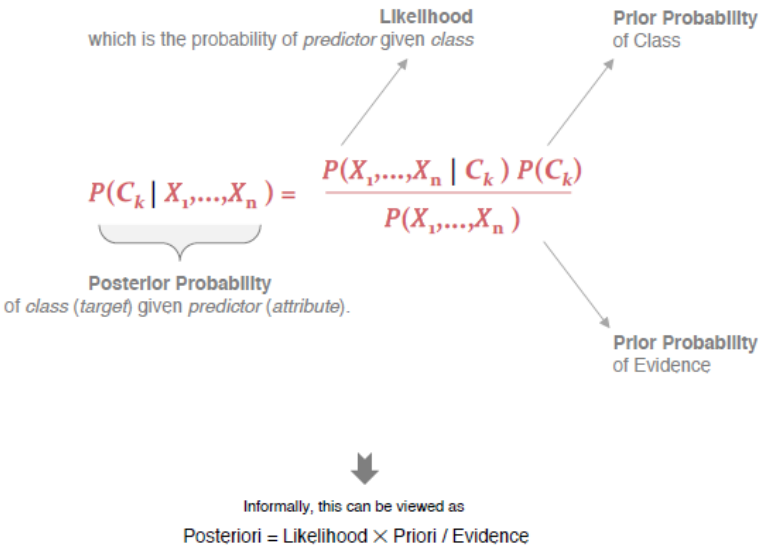
For each of the maneuver A1(turn left),A2(go straight),A3(turn right)… we can calculate their corresponding probability and rank them to find out the most probable predictions under fixed vehicle states history B.

Prediction Module

According to bayes rule, P(A|B) term is being called the Posteriori. which is proportional to LikelihoodP(B|A) and Prior Probability P(A).

$$P(A|B) \propto P(B|A) \times P(A)$$

P(B|A): ***the probability of predicted vehicle position and velocity(observations) given the turn maneuver class.***
P(A): ***the probability of a specific turn maneuver class.***



Finally, the equation to compute the probability of any exo-vehicle maneuver Mt can be written as:

$$P(Mt|Z0{:}t) \propto P(Zt|Mt) \times P(Mt)$$

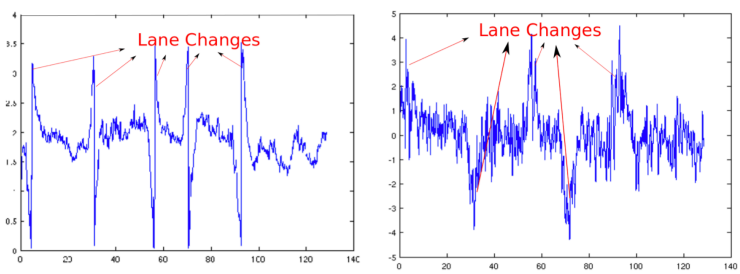
$$posterior \propto likelihood \times prior$$

The **Posterior** P(Mt|Z0:t) is to predict the probability of maneuver. where Z0:t corresponds to the historical observations between time 0 and time t.

The **likelihood** term P(Zt |Mt) , means the observability of a maneuver. The probabilistic output of a maneuver classifier (SVM algorithm) at each time step can directly be used as the likelihood.

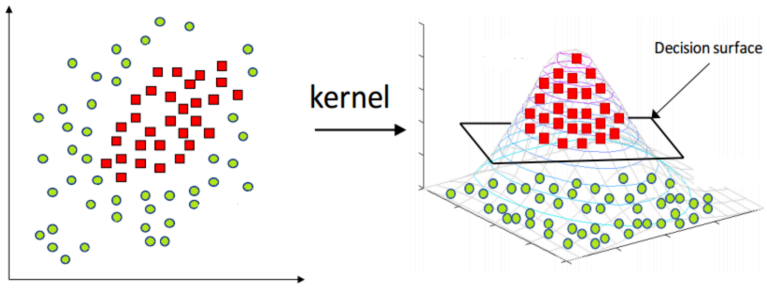
The SVM is a model off-line trained thru below procedures:

1. Data(e.g. road feature, vehicle status) collection for training
2. Designing suitable feature vector(e.g. lateral position, steering angle) correlated with maneuver class(e.g. lane changes).



Prediction Module

3. Kernel selection, The motivation behind using kernel method is to converting a nonlinear(complicated) classification problem at low dimension into a linear (easy) classification problem at high dimension.



4. Find best parameters using optimizing method

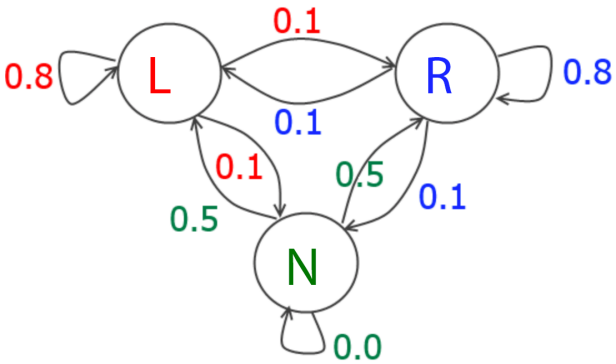
For a particular feature vector input at a particular instant, the SVM outputs the probability of this feature vector belonging to a particular class. This probability is **likelihood**.

The **Prior** term P(Mt) , is calculated by multiplying its former *Posterior* P(Mt-1|Z0:t-1) with *state transition probability* P(Mt|Mt-1):

$$[\sum_{M_{t-1}=L,R,N} P(M_t|M_{t-1}) \times P(M_{t-1}|Z_{0:t-1})]$$

Prior = Sum of (State Transition Probability × last Posterior)

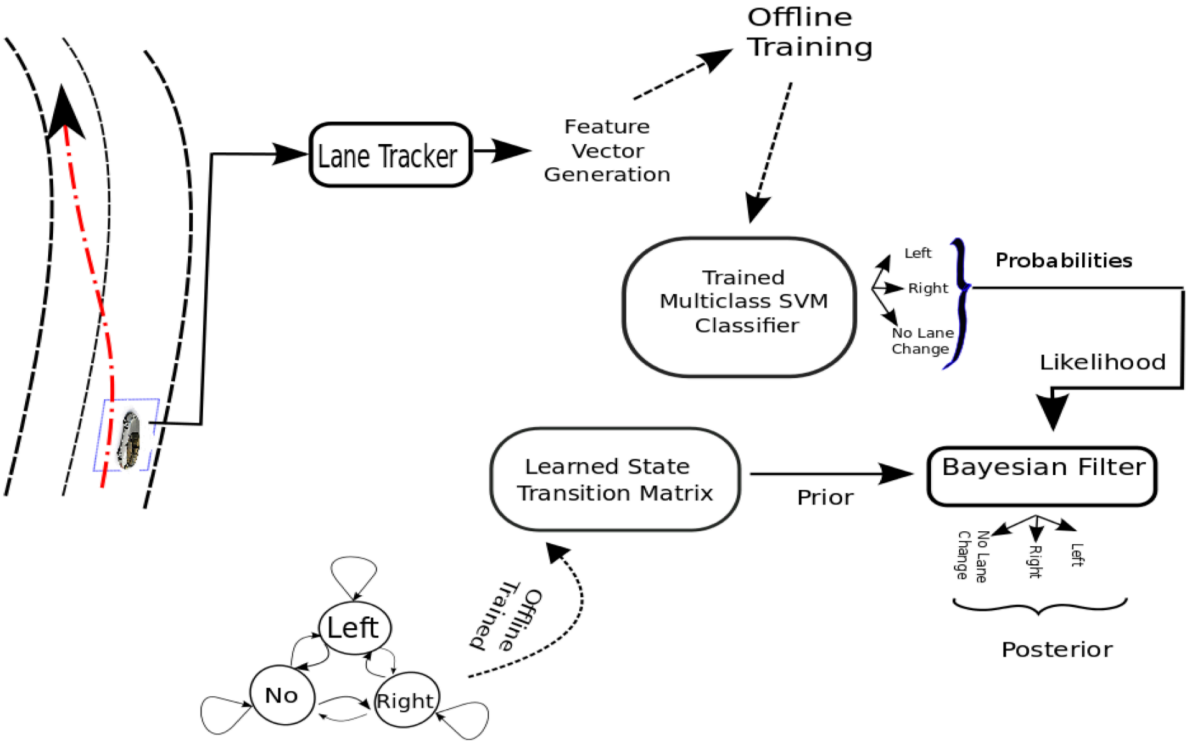
A demo (dummy value) state transition diagram for all the three states (L,R,N stands for left, right and no lane change) is shown in below graph, The state transition probabilities P(Mt |Mt –1) are learned offline using training data. For all the three maneuvers we get a 3 × 3 state transition matrix with 9 state transition probabilities.



$$\begin{array}{c} \\ L_{t-1} \\ R_{t-1} \\ N_{t-1} \end{array} \begin{array}{c} \begin{array}{ccc} L_t & R_t & N_t \end{array} \\ \begin{pmatrix} s_{11} & s_{12} & s_{13} \\ s_{21} & s_{22} & s_{23} \\ s_{31} & s_{32} & s_{33} \end{pmatrix} \end{array}$$

Prediction Module

Detailed overall approach: a **lane tracker** is used to generate lane features which is a prerequisite when searching for proper **feature vector** for later SVM. a generalized **maneuver class probability estimator (SVM)** is used to obtain probabilistic outputs. The **Bayesian method** takes the probabilistic output of the SVM as the **likelihood** input and the learned **state transition matrix** as the **prior** to find the **posterior** over all the possible classes. This **posterior** is finally used for the decision making.

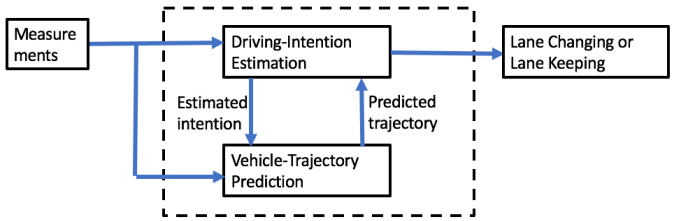


Prediction Module

Draw Lane-based trajectory

Now we will compute actual trajectory for an exo-vehicle if its driving intention is clarified.

indeed, Driving-intention estimation shall works with trajectory prediction together to decide if a lane changing or lane keeping maneuver would apply.

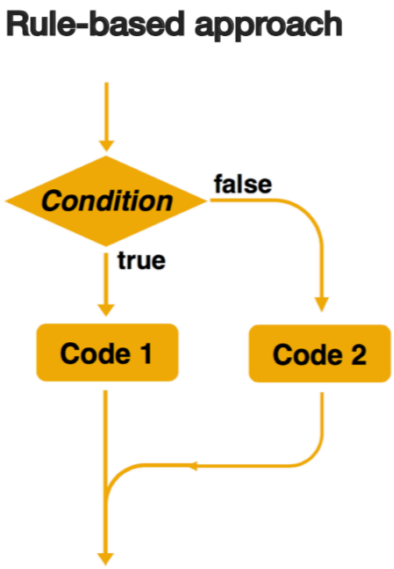


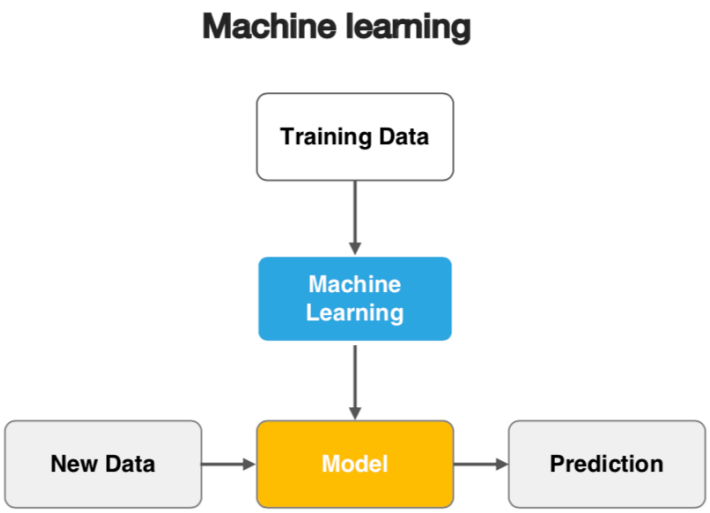
If the predicted trajectory collide with something detected by sensors, the predicted trajectory is initialized and re-planned then inform driving-intention estimator to update this change.

This strategy is quite similar like a human behavior which can be explained as the abortion of a lane change by a driver when he feels unsafe because of the insufficient gap or velocity. It is also expected to eliminate false alarms caused by zigzag driving.

Prediction Module

To predict trajectory, rule-based motion model and machine learning-based maneuver recognition model are frequently used.





Explicitly programmed to solve problem

Decision rules are clearly defined by humans

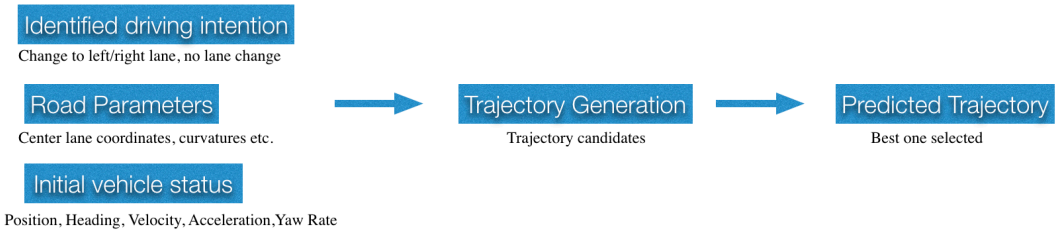
- Trained from examples
- Decision rules complex or fuzzy
- Rules are not defined by humans but learned by the machine from data

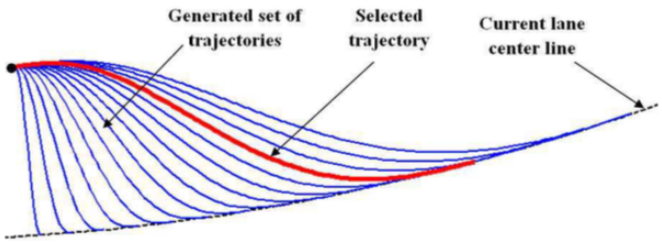
Machine learning approach is a behavior guessetimation of the future trend which is better to make long-term prediction whereas rule-based motion model is easy to interpret and generate decisions.

For the learning-based maneuver recognition model, there are many possible realizations for a single maneuver. Depending on the driver's habits, the actual trajectory may be pretty smooth or pretty aggressive. Moreover, the road geometry will also have an influence.

Prediction Module

So, based on the vehicle current state, the road parameters and the detected maneuver, a set of trajectories are first generated and the best one is selected with respect to a evaluating equation (cost function).

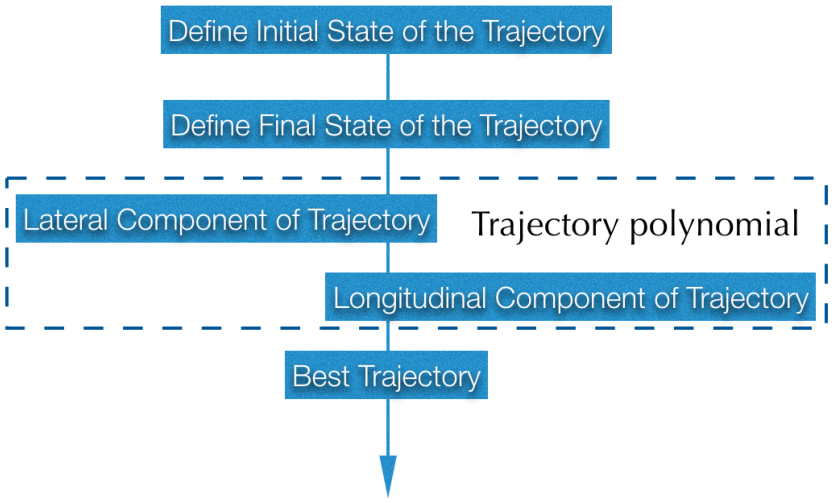




All the trajectories have the same initial state which is derived from the current exo-vehicle state(position, velocity, acceleration, etc.). at the end maneuver, the vehicle is moving right on the center line of its intended lane and has a constant longitudinal acceleration during the maneuver.

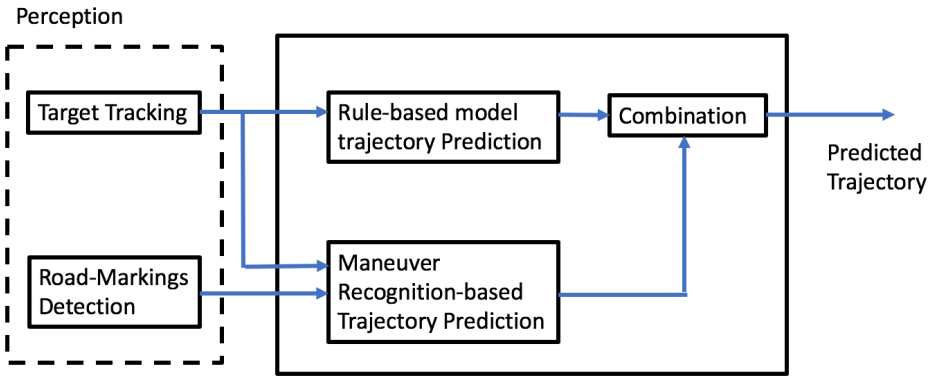
Series of trajectories will be modeled as a polynomial equation by using of the trajectory initial and final state to guaranties the jerk continuity (driving comfortable).

It is admitted that an average driver will seek to minimize the duration of his current maneuver but will also try to keep some comfort in the cockpit and avoid oscillation and overshoots. So best trajectory could be find according to those criteria mathematically.



Prediction Module

There are studies in which maneuver recognition can based on the comparison between target vehicle position and detected road center to replace the machine learning approach thus no more training data is needed.



Prediction Module

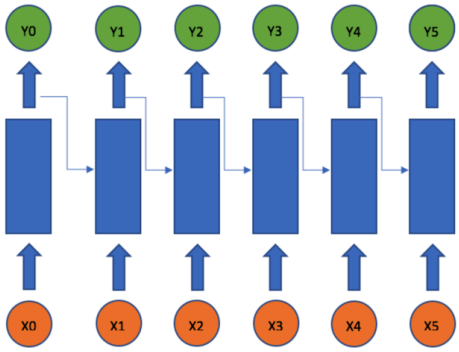
Recurrent Neural Networks

The simplest machine learning problem involving a sequence is a one to one problem. we have one data input to the model and the model generates a prediction with the given input.

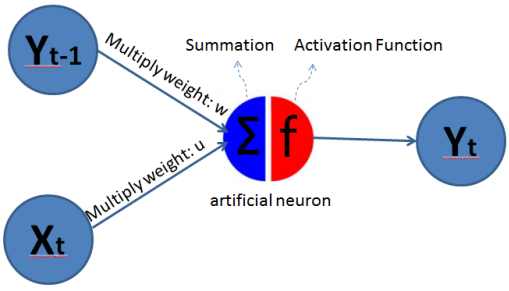
A recurrent neural network(RNN) could deals with sequence problems too.In below graph, we aligned several one-to-one network in parallel. But one more route attached to each adjacent networks: it retains the output state from the former network and use it as input for current network and so on to the next.

In programming terms this is like running a fixed program with certain inputs plus the retained output as its internal variables. This structure indicates that the network can remember previous input data, and uses this in combination with latest input data to make new predictions.

RNN retain information from last time series iteration, in a way it “remembers”. This memory is called the *state*.



In this graph case only two weights are involved. The weight multiplying the current input , which is *u,* and the weight multiplying the previous output *yt-1*, which is *w*.

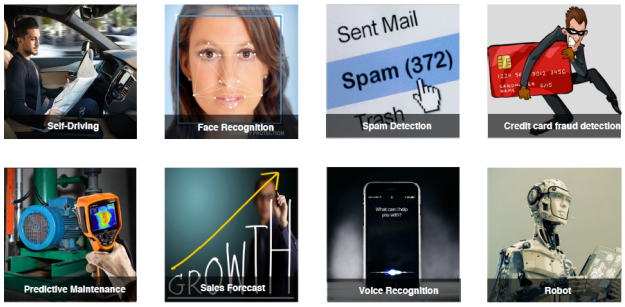


Prediction Module

Due to the recurrent nature, same weight parameters are used for all time series iterations.

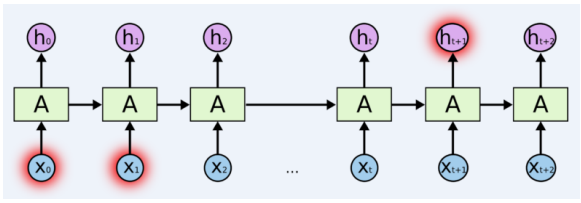
why is activation function used in this network? This is important because most real world data is non linear and we want neurons to learn these non linear representations. The activation function does the non-linear transformation to the input making it capable to learn and perform more complex tasks to simulate the real world.

in our case Hyperbolic tangent function(Tanh) is used to introduce the non-linear mechanism.

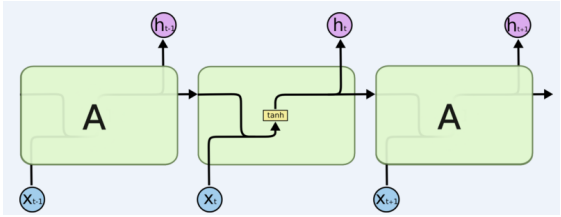


Variety tasks in real world.

A simple recurrent network suffers from a fundamental problem of not being able to capture long-term dependencies in a sequence. if time series get longer, RNN is not capable of linking remote infomations efficiently.

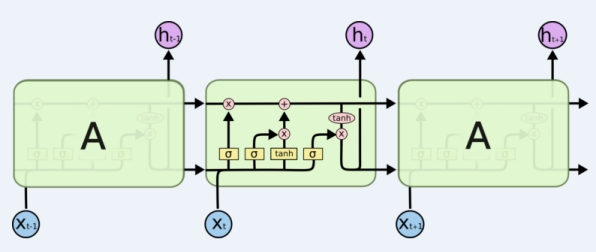


One popular variant of RNN is long short term memory (LSTM) model, which effectively overcomes the issue in naively designed RNNs.

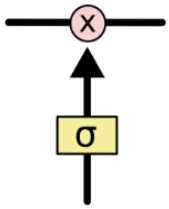


RNN model – only 1 activation function(sigmoid) applied.

Prediction Module

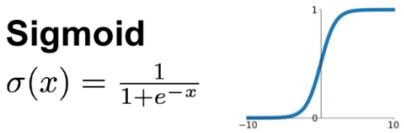


LSTM model – introduced 4 separate networks to each cell which makes it special



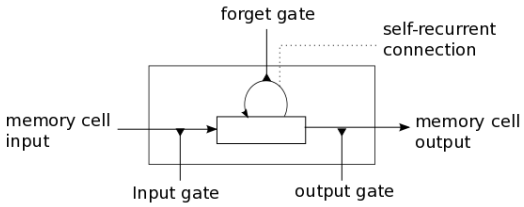
Gate Mechanism: sigmoid activation function plus a element multiplying operation

LSTM possess the ability to add or remove information in its element cell(like a memory). which ability is introduced by gate mechanism. gate selectively choose data to pass, sigmoid activation function yield a value between 0 and 1, 0 means allow the data to pass, and 0 is the opposite.

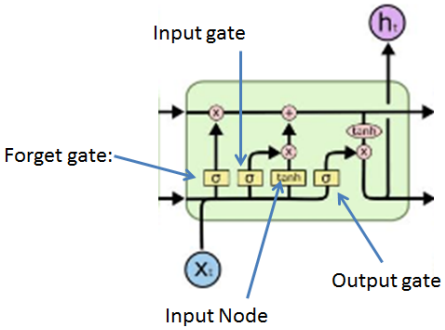


sigmoid function

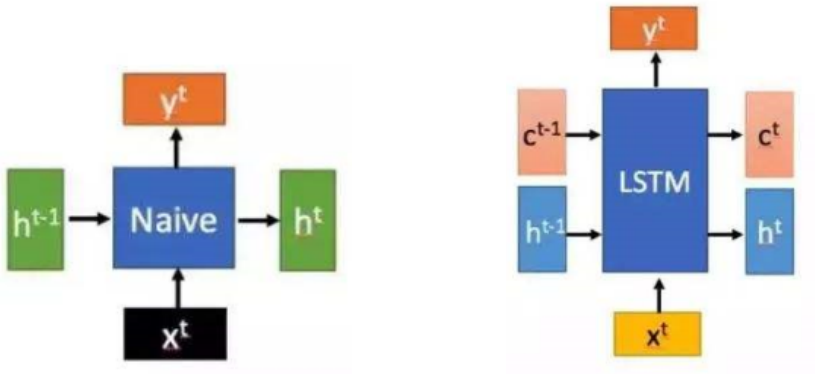
LSTM have 3 such gates to protect of control the in element cell. 3 gating mechanism(input gate,output gate and forget gate) ensures the information flow between the input, output, and cell memory are nicely controlled.



Prediction Module



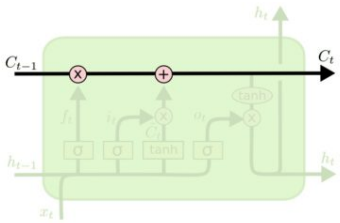
- ***forget gate*:** uses the inputs to decide how much to "forget" from the previous memory
- ***input gate*:** decides the amount of new information to be stored in memory
- ***output gate*:** computes the new cell output from a mix of the previous memory and new information stored in memory



Naive RNN LSTM

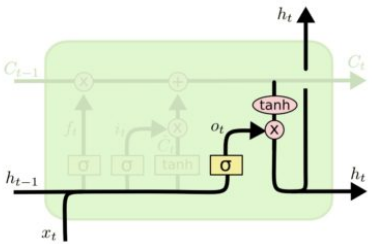
Naive RNN have only one trasmission state: Cell state. whereas the LSTM have 2 (Cell state and Hidden state)

Cell State is the accumulated state that gets updated(add something miner and forget something minor) from the previous time step hidden state.



Hidden State is calculated by applying the output gate to some linear combination of current cell state. Hidden State is also the output of current frame and input of next frame.

Prediction Module



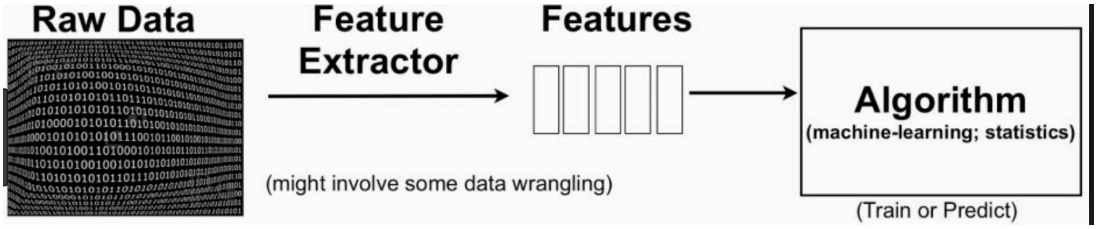
thus Cell state changes slowly whereas Hidden state changes faster.

This particular feature of LSTM allows a network to learn long-term relations between features, which makes them very powerful for time series prediction.

Prediction Module

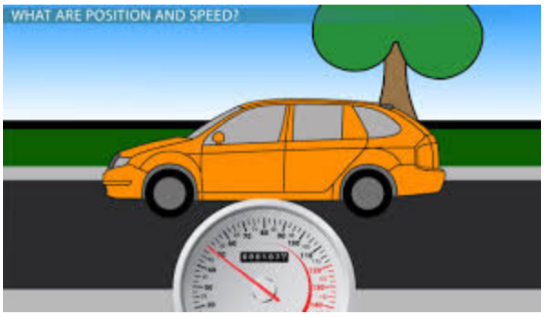
RNNs for Target Lane Prediction - Obstacle status feature and lane feature

Feature engineering and design is a key bridge to link raw data and vehicle behavior predictor (trajectory prediction algorithm). In our case, we could consider 2 categories of possible features: obstacle feature and lane feature.

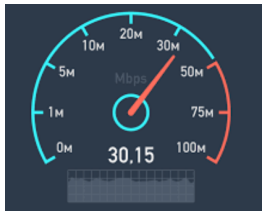


We aim at only using proper motion state features which can be reasonably easily measured using on-board sensors like LiDAR(laser) or Radar.

target tracking system(e.g. LiDAR, radar) hosted by the ego-vehicle provides for each target vehicle's position and velocity.



For prediction operation, we also need to know the ego-vehicle's state: velocity and yaw rate, provided by proprioceptive wheel speed sensor and gyroscopic device.



Prediction Module

Along the longitudinal direction of the target lane central reference, we can extract below lane features: center lane points global coordinates; lane curvature.

The estimated lane curvature is obtained by below equation, divide the ego vehicle's velocity into the ego-vehicle's yaw rate.

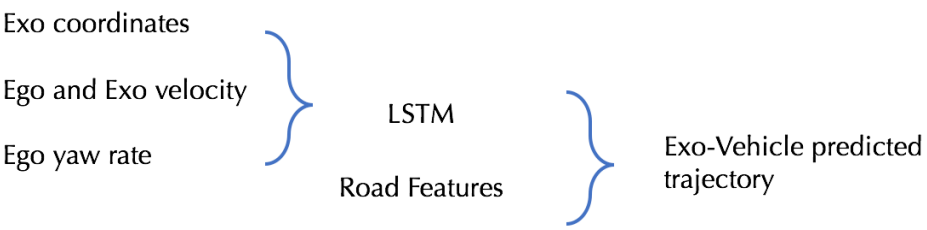
Prediction Module

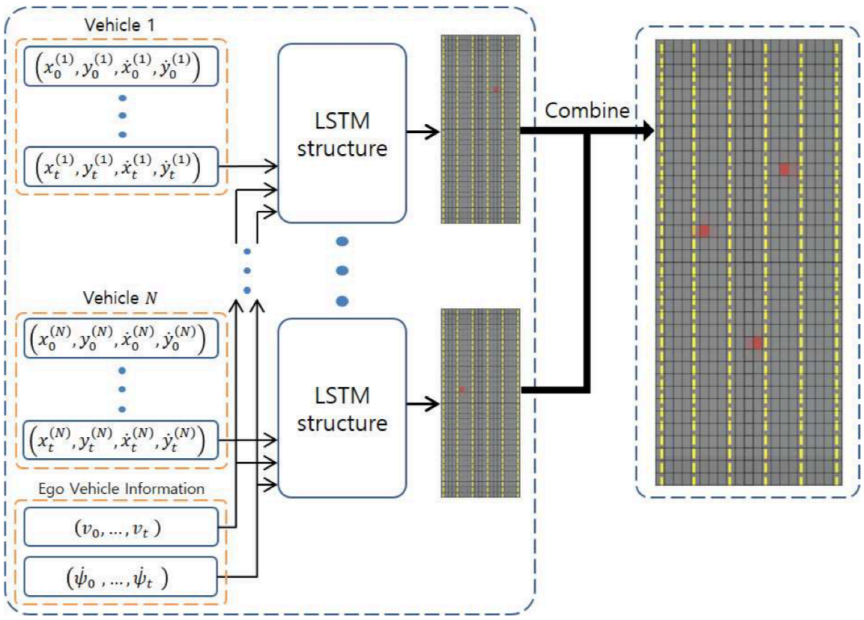
RNNs for Target Lane Prediction - Network Architecture

Our aim is to predict future positions for the target vehicle, we choose to use the LSTM(RNN) for our learning architecture, which is particularly well suited for time series problems.

Due to the recurrent nature, even a single layer of RNN nodes can iterate over time to form a "deep" neural network. the role of the RNN layer is to abstract a meaningful representation of the input time series then output the predicted future states. e.g. vehicle trajectory.

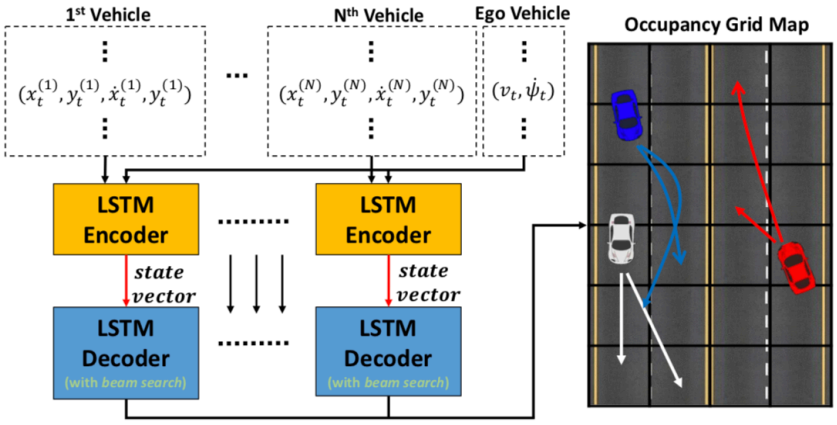
The proposed LSTM trajectory prediction system inputs the coordinates and velocities of the exo-vehicles and ego-vehicle's velocity and yaw rate obtained from the sensor measurements to the LSTM and produces the vehicle's future location after Δ seconds. The LSTM is designed to produce the probability of occupancy for the surrounding vehicles, plus the road features as input, Exo-vehicle's predicted trajectory is drafted.





Prediction Module

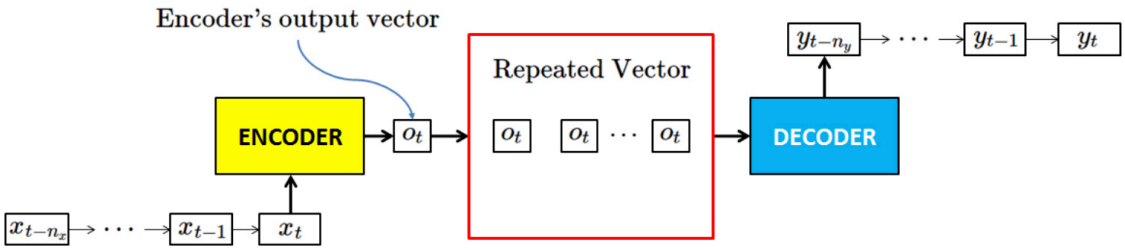
Recent studies introduced the LSTM encoder and decoder architecture for the trajectory prediction task.



It has a name: *Sequence to Sequence*. It means to put 2 LSTM networks in series.This structure is powerful as the size of the output is independent with the size of input data. i.e. both input and output are sequences but **with different length.**

How could it happen?

What the *encoder* actually did is creating a temporary output vector from the input sequence (you can think about that temporary output vector as a sequence with only one time step). Then, that vector is repeated n times, with n is the length of our desire output sequence.



First LSTM compress the sequence vehicle states to a context vector, which captures the key context of the past trajectory, Second LSTM generate future trajectory prediction from the context vector. actually we removed the physical limit about the length of the predicted output sequence.

Prediction Module

References

Comparative Evaluation of Occupancy Grid Mapping Methods Using Sonar Sensors	2:26 PM
Occupancy Grid Maps for Localization and Mapping	10:50 AM
Sequence to Sequence Learning with Neural Networks	10:37 AM
Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation	9:15 AM
DESIRE- Distant Future Prediction in Dynamic Scenes with Interacting Agents	9:12 AM
Probabilistic Vehicle Trajectory Prediction over Occupancy Grid Map via Recurrent Neural Network	9:10 AM
Deep Tracking- Seeing Beyond Seeing Using Recurrent Neural Networks	9:08 AM
Long short term memory 1997	9:04 AM
Analysis of Recurrent Neural Networks for Probabilistic Modeling of Driver Behavior	8:45 AM
Sequence-to-Sequence Prediction of Vehicle Trajectory via LSTM Encoder-Decoder Architecture	8:45 AM
MODELS FOR PEDESTRIAN TRAJECTORY PREDICTION AND NAVIGATION IN DYNAMIC ENVIRONMENTS	8:32 AM
PgglSCI14	12:51 AM
Predicting trajectories of golf balls using recurrent neural networks	12:45 AM
trajprediction	Yesterday
Vehicle Trajectory Prediction based on Motion Model and Maneuver Recognition	Yesterday
Models Supporting Trajectory Planning in Autonomous Vehicles	Yesterday
Vehicle_trajectories_prediction_based__on_driver_behaviour_model	Yesterday
Graphical Models for Driver Behavior Recognition in a SmartCar	Yesterday
Generalized Bradley-Terry Models and Multi-Class Probability Estimates	Yesterday
Using Support Vector Machines and Bayesian Filtering for Classifying Agent Intentions at Road Intersections	Yesterday
Behavior Classification Algorithms at Intersections and Validation using Naturalistic Data	Yesterday
Learning-based approach for online lane change intention prediction	Yesterday
A Lane Change Detection Approach using Feature Ranking with Maximized Predictive Power	Yesterday
ManeuverRecognitionOOBN_IITS2012	Yesterday
Car that Knows Before You Do- Anticipating Maneuvers via Learning Temporal Driving Models	Yesterday
dynamic bayesian networks. represetation,inference and learning	29/04/2018
probabilistic graphical models	29/04/2018
PATH AND TRAJECTORY PLANNING	29/04/2018
Sequence analysis of glance patterns to predict lane changes on urban arterial roads	29/04/2018
Probabilistic Driving Models and Lane Change Prediction	29/04/2018
Prediction of Pedestrian Trajectories Final Report	29/04/2018
Algorithms for Matching and Predicting Trajectories	29/04/2018
Target Lane Changing Prediction Method for ACC System	29/04/2018
Surround Vehicles Trajectory Analysis with Recurrent Neural Networks	29/04/2018
Multilayer Perceptron	29/04/2018
Predicting Destinations from Partial Trajectories Using Recurrent Neural Network	29/04/2018
Predicting trajectories of golf balls using recurrent neural networks ANTON JANSSON	28/04/2018
Modeling Trajectories with Recurrent Neural Networks	28/04/2018
Self-learning Trajectory Prediction with Recurrent Neural Networks at Intelligent Intersections	28/04/2018
Social LSTM Human Trajectory Prediction in Crowded Spaces	28/04/2018
Predicting Future Lane Changes of Other Highway Vehicles using RNN-based Deep Models	28/04/2018
Classification of Highway Lane Change Behavior to Detect Dangerous Cut-in Maneuvers	28/04/2018
Lane-Change Detection Based on Vehicle-Trajectory Prediction	28/04/2018
A Markov Model for Driver Turn Prediction	28/04/2018