

Stacked Recurrent Convolutional 모델을 이용한 주변 차량 Cut-in 의도 판단 알고리즘 개발

설재민, 최종기, 임세준*

국민대학교

snowjam02@kookmin.ac.kr, holysid@kookmin.ac.kr, *lim@kookmin.ac.kr

Development of a cut-in intention prediction algorithm for surrounding vehicles using the stacked recurrent convolutional model

Jaemin Seol, Jonggi Choi, Sejoon Lim*

Kookmin Univ.

요 약

본 논문에서는 Vissim 교통 시뮬레이터를 사용하여 여러 Cut-in 상황에서 추출한 데이터를 사용해 Cut-in 의도를 예측하였다. 학습에는 LSTM(Long Short Term Memory)과 3D-CNN(3 Dimensional-Convolution Neural Network)을 결합한 SRC(Stacked Recurrent Convolutional) 모델을 사용하였다. 관측된 경로와 LSTM으로 예측한 경로를 3D-CNN의 입력으로 넣었으며 LSTM의 예측범위와 3D-CNN의 하이퍼 파라미터를 수정하며 비교하여 95% 이상의 높은 예측 정확도로 Cut-in 의도를 예측하였다.

I. 서 론

전세계적으로 매년 약 125만명이 교통사고로 사망하며 이 중 운전자의 인적요인으로 인한 사망이 90%에 달한다.[1] 미국에서는 운전자의 인적요인으로 인한 차선변경 사고가 매년 25만건 발생한다.[2] 자율주행 차량은 인간이므로 발생할 수 있는 인적오류를 최소화하여 사고 발생 가능성을 낮추며, 과속과 같은 교통법규위반을 해소할 방안이 될 수 있다.

아직 Society of Automotive Engineers에서 정한 자율주행의 6단계 중 Level 5 (완전 자율 주행)[3]이 적용된 차량이 실제 주행을 허가받지는 못했지만, 완전 자율주행 차량을 위한 연구는 다양한 분야에서 진행되고 있다. 자율주행 차량의 가장 큰 이슈는 안전성인데 본 논문에서는 그 중 주변 차량의 Cut-in 의도를 예측하여 차량의 충돌위험을 줄이기 위한 연구를 진행하였다.

UC버클리의 연구팀은 실제 운전자의 행동방식을 모사하여 움직이고 주변 차량의 행동을 예측하는 방법을 사용하였고[4] 현대자동차는 주변 차량들의 크기, 위치, 이탈각 등의 데이터를 바탕으로 주변 차량의 거동을 예측하여 안전한 경로를 탐색하였다.[5] 위에 서술한 연구들은 rule-based 방식을 사용하여 주변차량의 행동을 예측하였다.

미시건 대학에서는 RNN을 기반으로 하는 LSTM(Long Short-Term Memory)과 SRNN(Structural-RNN)을 사용하여 주변 차량의 Cut-in 의도를 검출하였다. 이 연구는 3차선, 차차 주변 6대의 차량의 움직임을 파악하여 Cut-in을 40%정도의 정확도로 예측하였다. [6]

SRC 모델은 예측 정확도가 높은 LSTM(Long Short-Term Memory)을 사용하여 미래의 차차 및 주변 차량의 경로와 heading angel 값을 예측한 후, 기존의 시계열 데이터와 예측한 시계열 데이터를 합쳐 3D-CNN의 입력 데이터로 사용한다. 차차 기준 총 9개의 분면으로 나뉜 중앙을 자차의 위치로 정의한 후 각 분면의 위치에 해당하는 차차와 거리가 가장 가까운 차량들을 감지해 차차와의 상호관계를 정의하는 4D 형태의 vehicle data buffer를 정의한다. 4D 형태의 data buffer에서 차량의 추상화된 위치에 대한 x, y 축과 시간 축으로 이동하는 filter로 3D-CNN을 사용하여 차차

와 주변 차량 사이의 관계를 추출하여 Cut-in 의도를 예측하는 알고리즘을 개발하였으며 시뮬레이션을 이용하여 검증하였다.

II. 문제 정의

본 논문은 임의의 차량에 대한 주변 차량들의 Cut-in 의도를 예측한다. 과거 시간 T_{past} 에서 현재 시간 T_{now} 까지 시간 t에 존재하는 임의의 ego 차량의 좌표와 target 차량의 좌표를 각각 $(x_E[t], y_E[t])$, $(x_T[t], y_T[t])$, 시간 t에서의 ego 차량과 target 차량의 heading angle값을 각각 $h_E[t]$, $h_T[t]$, sampling rate를 fHz라 정의한다. $n = (T_{now} - T_{past}) \cdot f$ 일 때, 차량의 위치는 다음과 같다.

$$x_E[i] = (x_E[T_{past} + \frac{1}{f} \cdot i] - x_E[T_{past}]) \quad (i = 0, 1, 2, \dots, n) \quad (1)$$

$$y_E[i] = (y_E[T_{past} + \frac{1}{f} \cdot i] - y_E[T_{past}]) \quad (i = 0, 1, 2, \dots, n) \quad (2)$$

$$x_T[i] = (x_T[T_{past} + \frac{1}{f} \cdot i]) \quad (i = 0, 1, 2, \dots, n) \quad (3)$$

$$y_T[i] = (y_T[T_{past} + \frac{1}{f} \cdot i]) \quad (i = 0, 1, 2, \dots, n) \quad (4)$$

위의 식 (1)~(4)를 이용하여 target 차량의 위치를 ego 차량에 대한 상대 거리로 나타내면 다음과 같이 정의할 수 있다.

$$x_T^*[i] = (x_T[i] - x_E[i] - x_E[T_{past}]) \quad (i = 0, 1, 2, \dots, n) \quad (5)$$

$$y_T^*[i] = (y_T[i] - y_E[i] - y_E[T_{past}]) \quad (i = 0, 1, 2, \dots, n) \quad (6)$$

즉, 시간 t에서의 ego 차량의 데이터는 $[x_E[t], y_E[t], h_E[t]]$, target 차량의 데이터는 $[x_T^*[t], y_T^*[t], h_T[t]]$ 로 표현 가능하다. 해당 데이터를 시계열로 쌓아 올려 주변 차량의 Cut-in 의도를 판단한다. 주변 차량의 Cut-in 의도를 판단하기 위해 Cut-in 상황에 대해 label 0 ~ 12의 13개 라벨로 나눈다. 해당 라벨에 대한 설명은 [III. (2) 차량의 위치 정보 추상화 및 라벨링]에서 하도록 하며 각각의 label값은 13까지의 one hot encoding 된 값이고 [그림 1]과 같다.

label 0 : [1 0 0 0 0 0 0 0 0 0 0 0 0]
 label 1 : [0 1 0 0 0 0 0 0 0 0 0 0 0]
 ⋮
 label 12 : [0 0 0 0 0 0 0 0 0 0 0 0 1]

[그림 1] One-hot encoding된 라벨값

III. 알고리즘

(1) Stacked Recurrent Convolutional 모델 학습 알고리즘

본 논문에서는 Stacked Recurrent Convolutional Model을 고안하였다. SRC model의 기본적인 구조는 [그림 2]와 같다. LSTM(Long Short Term Memory)을 사용하여 target 차량의 Cut-in 의도 발생 후 T 초간의 x,y, heading angle값을 이용하여 직후의 n 초의 시간에 대해 해당 값들을 예측하게 학습시킨다. LSTM 모델로 학습을 완료한 뒤 예측한 값을 기존 학습 데이터의 T 초 뒤에 추가하여 $T+n$ 초 더 긴 시계열 데이터를 생성한다. 생성한 $T+n$ 초의 길이를 가진 시계열 데이터를 입력 데이터로 사용하여 3D CNN(3 Dimensional Convolution Neural Network) 모델의 입력 데이터로 사용해 주변 차량과의 관계를 학습하여 13개의 라벨값에 대해 예측을 하는 모델을 생성하여 학습시킨다.



[그림 2] Recurrent Stacked Ensemble 구조

(2) 차량의 위치 정보 추상화 및 라벨링

ego 차량의 x좌표가 E_x 일때, [그림 3]과 같이 $-\frac{100}{3}(m) \leq E_x < \frac{100}{3}(m)$ 인 ego 차량의 차선을 구역 5로 정의하며 진행 방향 기준 왼쪽 차선을 구역 4, 오른쪽 차선을 구역 6으로 정의한다. target 차량의 x좌표를 T_x 라 정의할 때 $\frac{100}{3}(m) \leq (T_x - E_x) < 100(m)$ 에 해당하는 공간에 대해 ego 차량의 왼쪽 차선에 존재할 경우 구역 1, 같은 차선은 구역 2, 오른쪽 차선은 구역 3에 차량이 존재한다고 추상화시킨다. 이와 같은 방법으로 $-100(m) \leq (T_x - E_x) < -\frac{100}{3}(m)$ 의 공간에 대해 ego 차선의 왼쪽 차선은 구역 7, 같은 차선은 구역 8, 오른쪽 차선은 구역 9로 정의한다. 추상화한 구역을 기반으로 직진하는 차량들에 대해서는 label 0으로 정의하고 1번 구역에 대해 target 차량이 앞으로 Cut-in하는 상황이 label 1, 뒤로 Cut-in하는 상황이 label 7, 3번 구역에 대해 target 차량이 앞으로 Cut-in하는 상황이 label 2, 뒤로 Cut-in하는 상황이 label 8로 정의하였다. 이와 마찬가지로 4번 구역에 대해 label 3, label 9, 6번 구역에 대해 label 4, label 10, 7번 구역에 대해 label 5, label 11, 9번 구역에 대해 label 6, label 12로 정의하였다.

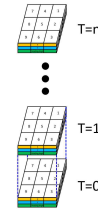


[그림 3] 차량 위치 정보의 추상화

(3) 위치 정보 추상화가 적용된 데이터 버퍼

ego 차량과 target 차량의 위치가 어디에 있는 [그림 3]과 같이 위치 정보를 추상화하여 표현할 수 있다. [그림 4]에서와 같이 추상화된 구역에 해당하는 target 차량의 위치에 $[x_T^*[t], y_T^*[t], h_T[t]]$ 데이터를, ego 차량

데이터 $[x_E[t], y_E[t], h_E[t]]$ 를 구역 5에 추가하여 시간순으로 쌓아 $n \times 3 \times 3 \times 3$ 의 모양을 가진 4D 버퍼를 제작한다.



[그림 4] 위치 추상화를 이용한 데이터 버퍼

(4) 3D CNN 학습 알고리즘

ego 차량과 target 차량의 fHz의 sampling rate을 가진 데이터의 T 초간의 이동 경로와 heading angle값을 이용하여 $T \cdot f \times 3 \times 3 \times 3$ 의 데이터 버퍼를 이용해 3D CNN 모델을 학습시킨다. 해당 모델은 두 개의 convolution layer를 지나며 ego 차량과 target 차량 사이의 관계에 대해 추출하며 학습한 뒤 두 개의 Fully Connected Layer를 통해 학습한 관계에 대해 판별을 하며 Cut-in 의도에 대한 예측을 한다. 이 때, 오차값은 예측한 label에 대해 softmax값을 구해 Cross Entropy를 산출하여 사용하는 Categorical Cross Entropy를 사용해 해당 오차값을 줄이는 방향으로 학습을 진행하였다. softmax값을 구하는 식 (7)과 Cross Entropy에 대한 식 (8)은 다음과 같다.

$$f(x)_i = \frac{e^{s_i}}{\sum_j e^{s_j}} \quad (T \text{는 전체 label 개수}) \quad (7)$$

$$\text{Cross Entropy} = - \sum_i v_i \cdot \log(f(x)_i) \quad (v_i \text{는 원핫 인코딩 된 라벨값}) \quad (8)$$

(5) LSTM 학습 알고리즘

LSTM을 이용하여 식 예측 시각 T_n 을 기준으로 T_p 부터의 입력 데이터를 사용하여 k초간의 차량의 이동 경로 및 heading angle을 예측한다. 식(1)과 식(2)를 이용하여 fHz의 sampling rate을 가진 데이터에서 T_k 초간의 데이터를 뽑아 $[x_E[0], y_E[0], h_E[0]] \sim [x_E[T_k], y_E[T_k], h_E[T_k]]$ 까지 쌓아올려 $T_k \cdot f \times 3$ 의 모양을 가진 2차원 배열을 생성한다. 마찬가지로 식(5)와 식(6)을 이용하여 만든 target 차량에 대한 데이터를 쌓아 $[x_T^*[0], y_T^*[0], h_T^*[0]] \sim [x_T^*[T_k \cdot f], y_T^*[T_k \cdot f], h_T^*[T_k \cdot f]]$ 인, $(T_k \cdot f + 1) \times 3$ 의 모양의 2차원 배열을 생성한다. target 차량의 Cut-in 의도 생성 후 T_l 초 동안의 데이터를 사용하여 해당 시간 동안의 데이터를 $(T_l - T_k) \cdot f \times 3$ 형태의 LSTM 모델 예측 결과값으로 생성한다. 생성한 데이터를 바탕으로 값을 예측하는 LSTM 모델을 생성하여 학습시킨다. 학습 진행 시 오차값은 예측하는 시간에 해당하는 각각의 x 값, y 값, heading angle값에 대해 Mean Squared Error(MSE)를 산출하여 해당 오차값을 줄이는 방향으로 학습을 진행하였다. MSE에 대한 식은 아래와 같다.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - t_i)^2 \quad (9)$$

IV. 모델 학습 및 결과

(1) 모델 학습을 위한 데이터

본 논문의 모델에 대한 학습은 PTV(Planung Transport Verkehr AG)사에서 제작한 PTV VISSIM 시뮬레이터를 사용하였다. 해당 시뮬레이터

를 사용하여 직선 3차선의 고속도로 상황을 생성하여 데이터를 추출 및 전처리하였다. PTV VISSIM 시뮬레이터를 사용하여 추출한 데이터를 전처리하여, Training set과 Validation set, Test set에 대해 각각 60%, 20%, 20%로 맞춰 학습하는 데 사용하였다.

(2) LSTM 모델 학습

LSTM 모델은 예측하려는 구간이 0.5초와 1초 두 개임에 따라 각각 LSTM1모델과 LSTM2 모델 두가지로 나누어 제작 및 학습을 진행하였다. LSTM1 모델과 LSTM2 모델 모두 PTV VISSIM 차량 데이터를 사용하여 학습을 진행하였다.

(3) 3D CNN 모델 학습

3D CNN 모델은 LSTM 모델이 2개임에 따라 LSTM 모델을 사용하지 않는 3D CNN 모델(총 1초의 입력 데이터)4개와 0.5초를 예측하는 LSTM1 모델과 결합한 3D CNN 모델(총 1.5초의 입력 데이터)4개, 1초 예측을 하는 LSTM2 모델(총 2초의 입력 데이터)을 이용한 3D CNN모델 4개로 총 12개의 모델을 제작하여 학습을 진행하였다. 각각의 모델들은 학습하는 커널 크기에 따라 나누었다.

(4) 결과

학습 결과 train set에 대해 LSTM을 쓰지 않은 모델에서는 평균적으로 0.8684의 정확도를 얻었고 0.5초를 예측하는 LSTM을 사용한 모델은 0.9714, 1초를 예측하는 LSTM 모델은 0.9683의 정확도를 얻었다. 각 모델의 예측 정확도는 다음 [표 1]과 같다.

커널 사이즈 LSTM모델	(3,2,2)	(5,2,2)	(3,3,3)	(5,3,3)	평균 예측 정확도
No LSTM	0.8997	0.8402	0.8404	0.8934	0.8684
LSTM1	0.9920	0.9458	0.9739	0.9848	0.9714
LSTM2	0.9915	0.9730	0.9215	0.9871	0.9683
평균 예측 정확도	0.9611	0.9197	0.9119	0.9551	

[표 1] 각 모델에 따른 test set에 대한 예측 정확도

[표 1]을 보면 LSTM을 사용한 모델들이 LSTM을 사용하지 않은 모델에 비해 test set에 대한 정답을 맞히는 정확도가 최소 8%이상 높은 것을 확인할 수 있다. LSTM을 사용하여 예측할 경우 정확도의 측면뿐 아니라 위험 상황에 대해 더 빠른 예측이 가능해진다. 즉, 차량의 Cut-in이 발생하기 1초전 차량의 Cut-in을 예측하는 것이 아니라 Cut-in 발생 1.5초전, 2초전에 차량에 대한 Cut-in 의도를 예측하는 것이 가능해진다.

V. 결론

본 논문에서 LSTM과 3D CNN을 활용하여 주변차량의 Cut-in 의도를 예측하는 모델을 제시하였으며, VISSIM 시뮬레이터를 활용하여 모델을 학습하고 검증하였다. 높은 성능의 Cut-in 상황 예측은 교통사고의 가능성을 크게 줄여 원활한 도로 상황을 유지할 수 있게 한다. 해당 연구를 바탕으로 시뮬레이션을 통한 모델의 Cut-in 예측 성능을 고도화하는 연구를 진행하고 이를 바탕으로 실제 차량을 이용한 연구를 진행할 계획이다.

ACKNOWLEDGMENT

이 연구는 2020년도 산업통상자원부 ‘자동차산업핵심기술개발사업’의 재원으로 산업기술평가관리원(KEIT) 연구비 지원(10083646) 및 산업통상자원부 ‘산업전문인력역량강화사업’의 재원으로 한국산업기술진흥원(KIAT)의

지원(2020년 미래형자동차 R&D 전문인력 양성사업, 과제번호: N0002428)을 받아 수행된 연구임.

참 고 문 헌

- [1] Global status report on road safety 2018 P.22, World Health Organization
- [2] Singh, S. (2015). Critical reasons for crashes investigated in the national motor vehicle crash causation survey (NO. DOT HS 812 115).
- [3] 김중효, 허낙원, 김태경, "자율주행 AI 운전능력 평가기법 및 모형개발에 관한 연구", 도로교통공단 교통과학연구원, 2018
- [4] Sadigh, Dorsa, et al., "Information gathering actions over human internal state." Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on. IEEE, 2016.
- [5] 유수정, 윤지현, 신민용, 김희원 (현대자동차), "차량의 자율 주행 시스템 및 방법", 특허출원 KR101610544B1, 2014.
- [6] Predicting Future Lane Changes of Other Highway Vehicles using RNN-based Deep Models(2018) - Sajan Patel, Brent Griffin, Kristofer Kusano, and Jason J. Corso