
자율주행이동로봇

기말과제



제출일	2023. 12. 19	전공	로봇학과
과목	자율주행이동로봇	학번	1758028
담당교수	정세영	이름	정성훈

학습 데이터 - 435개

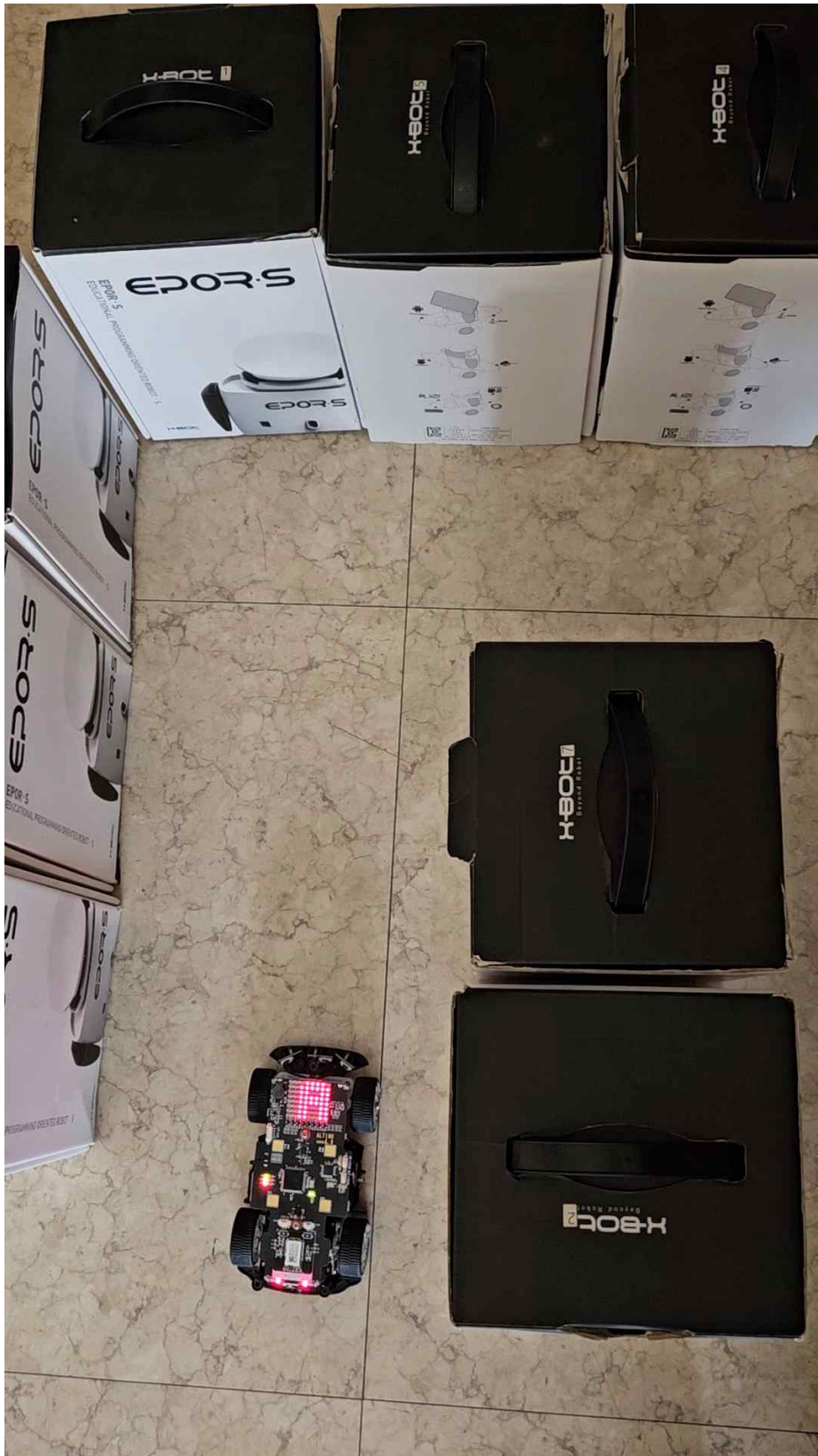
Left	Front	Right		Forward	Steering
0	0	7	300	300	0
0	0	5	300	300	0
0	0	1	300	300	0
0	0	2	300	300	0
0	0	2	300	300	-127
0	0	3	300	300	-127
0	0	6	300	300	-127
0	0	7	300	300	-127
0	0	7	300	300	0
0	0	0	300	300	0
0	0	1	300	300	0
0	0	0	300	300	0
0	0	1	300	300	0
0	0	1	300	300	127
0	0	0	300	300	127
0	0	4	300	300	127
0	0	0	300	300	127
0	0	1	300	300	127
0	0	0	300	300	127
4	0	2	300	300	127
13	4	1	300	300	127
16	6	0	300	300	127
20	8	2	300	300	127
20	8	2	300	300	0
26	7	0	300	300	0
26	7	0	0	0	0
32	9	0	0	0	0
44	13	0	0	0	0
65	23	1	0	0	0

학습 시킬 때 출력 결과(조향 및 전후진)을 -1,0,1로 통일 함.

학습 시킬 때 오차율이 300,127 등으로 하니 크게 나와 학습 정도 파악이 어려움.

이후 사용할 때 다시 300,127등으로 다시 올려서 사용.

학습에 사용된 트랙



전진 학습

```
model = Sequential()
model.add(Dense(100, activation='relu', input_dim=3))
model.add(Dense(1000, activation='relu'))
|
model.add(Dense(1))
```

```
sgd=optimizers.Nadam(learning_rate=0.001)
model.compile(optimizer=sgd, loss='mean_squared_error')
model.summary()
history = model.fit(x_train, y_train, epochs=5000, batch_size=256)
```

Model: "sequential_23"

Layer (type)	Output Shape	Param #
dense_95 (Dense)	(None, 100)	400
dense_96 (Dense)	(None, 1000)	101000
dense_97 (Dense)	(None, 1)	1001

Total params: 102,401

Trainable params: 102,401

Non-trainable params: 0

Epoch 1/5000

3/3 [=====] - 0s 4ms/step - loss: 0.1013

Epoch 2/5000

3/3 [=====] - 0s 8ms/step - loss: 0.1026

Epoch 3/5000

3/3 [=====] - 0s 6ms/step - loss: 0.0948

Epoch 4/5000

3/3 [=====] - 0s 6ms/step - loss: 0.0871

Epoch 5/5000

3/3 [=====] - 0s 7ms/step - loss: 0.0899

...

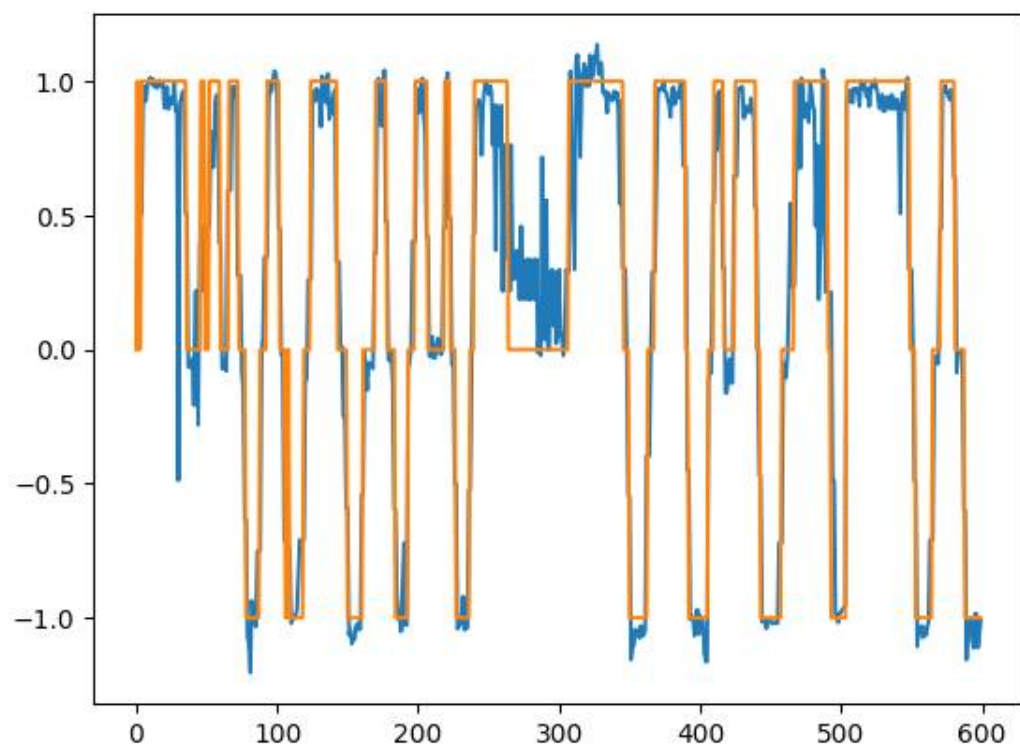
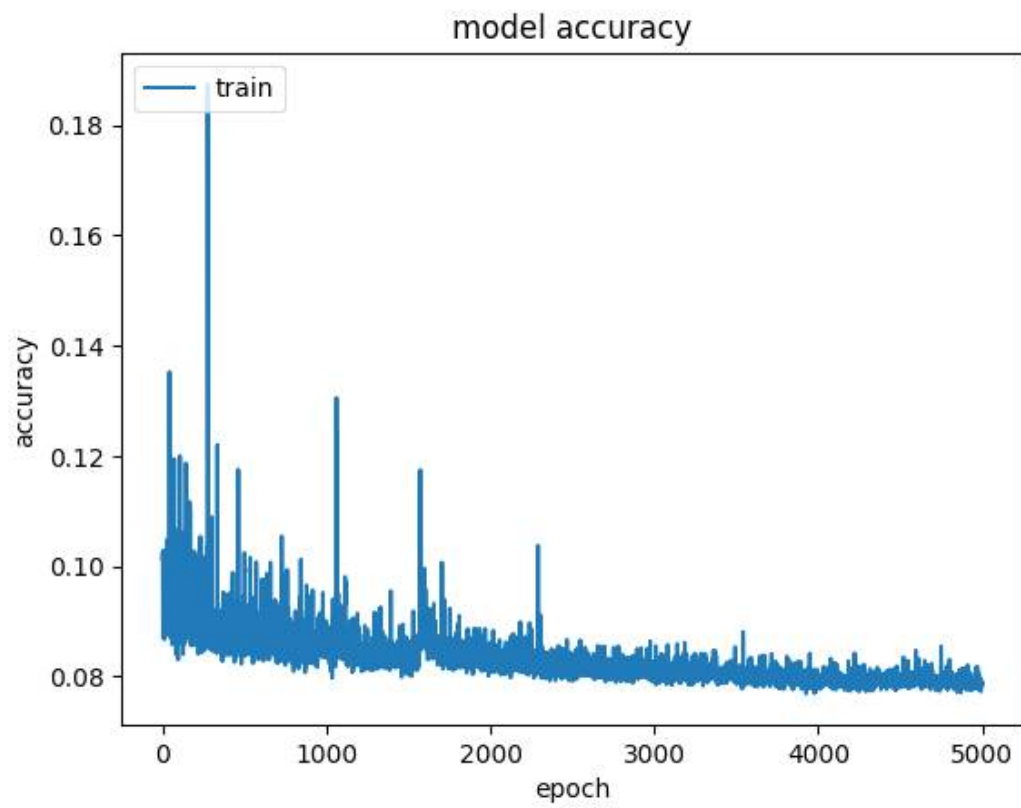
Epoch 4999/5000

3/3 [=====] - 0s 4ms/step - loss: 0.0791

Epoch 5000/5000

3/3 [=====] - 0s 4ms/step - loss: 0.0784

전진 학습 결과



스티어링 학습

```
model = Sequential()
model.add(Dense(100, activation='relu', input_dim=3))
model.add(Dense(1000, activation='relu'))
model.add(Dense(200, activation='sigmoid'))
model.add(Dense(500, activation='relu'))
model.add(Dense(300, activation='tanh'))
model.add(Dense(100, activation='relu'))

model.add(Dense(1))

sgd=optimizers.Nadam(learning_rate=0.001)
model.compile(optimizer=sgd, loss='mean_squared_error')
model.summary()
history = model.fit(x_train, y_train, epochs=5000, batch_size=256)
```

Model: "sequential_9"

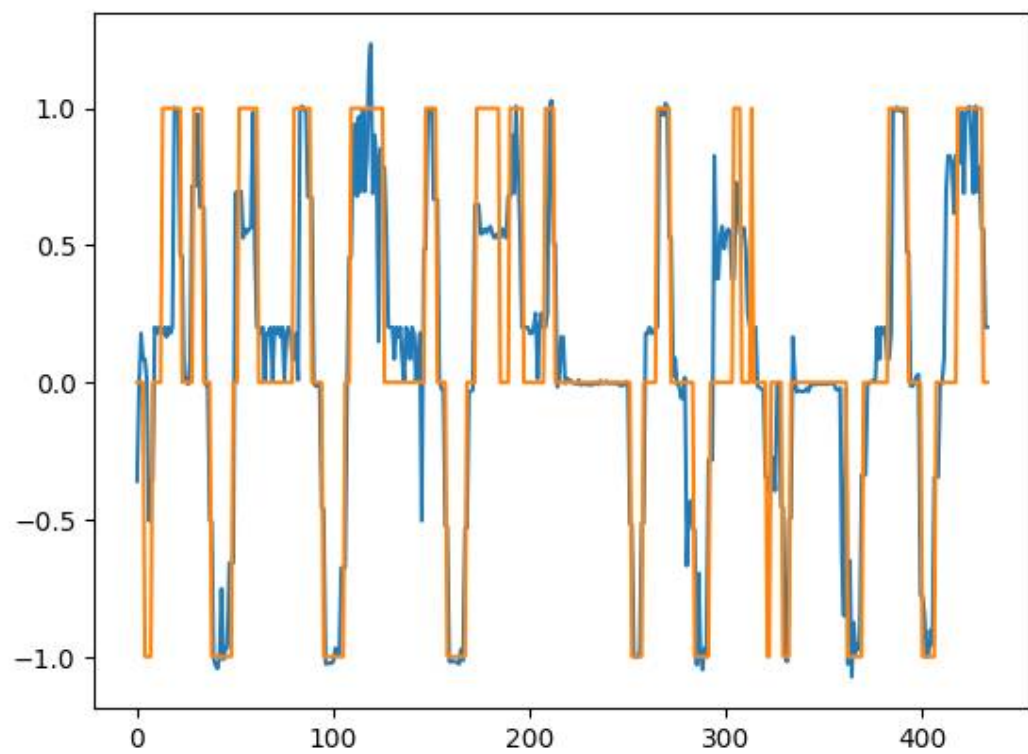
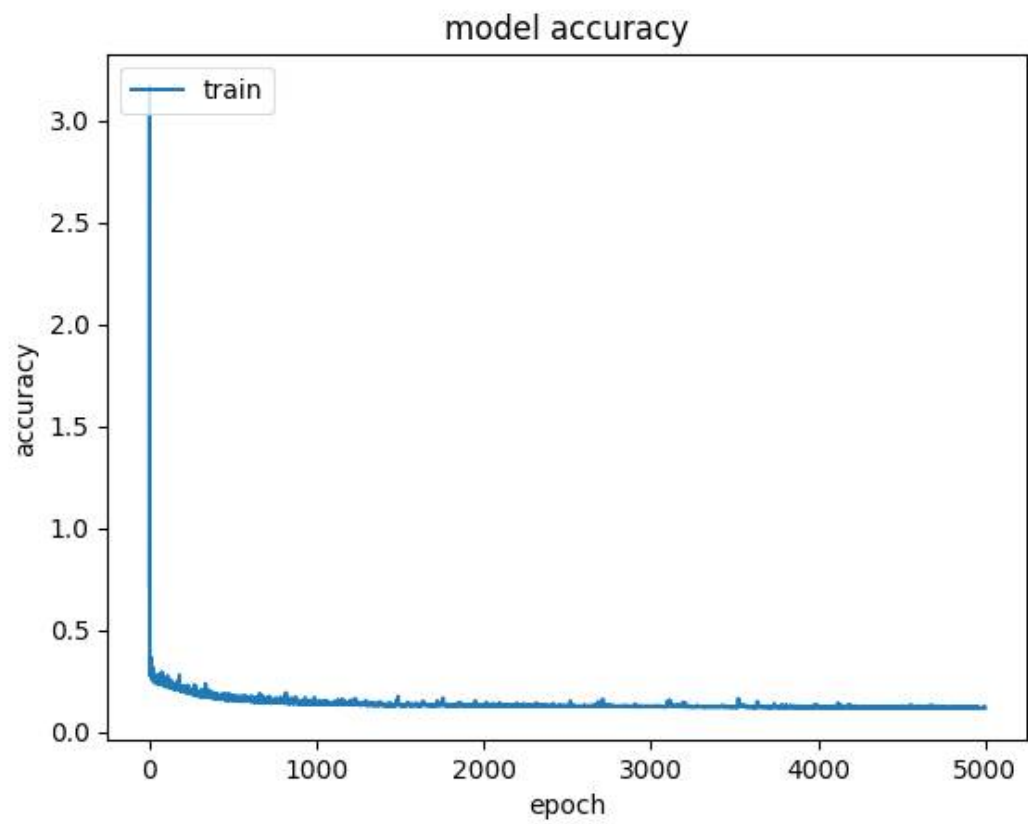
Layer (type)	Output Shape	Param #
dense_51 (Dense)	(None, 100)	400
dense_52 (Dense)	(None, 1000)	101000
dense_53 (Dense)	(None, 200)	200200
dense_54 (Dense)	(None, 500)	100500
dense_55 (Dense)	(None, 300)	150300
dense_56 (Dense)	(None, 100)	30100
dense_57 (Dense)	(None, 1)	101

Total params: 582,601
Trainable params: 582,601
Non-trainable params: 0

Epoch 1/5000
2/2 [=====] - 1s 15ms/step - loss: 0.7160
...
Epoch 4999/5000
2/2 [=====] - 0s 5ms/step - loss: 0.1146
Epoch 5000/5000
2/2 [=====] - 0s 7ms/step - loss: 0.1178

Output is truncated. View as a [scrollable element](#) or open in a [text editor](#). Adjust cell output [settings...](#)

스티어링 학습 결과



자율 주행 코드

```
S_model =load_model('Steer_R.h5')
F_model =load_model('Foward_R.h5')
#print(S_model.predict(np.array([[0, 0, 50]])))
Open()
num =1
result =0
while 1:

    if keyboard.is_pressed("q"):
        break
        S_result      =S_model.predict(np.array([[sensor.IR[1],
sensor.IR[2], sensor.IR[3]]]),verbose=0)
        F_result      =F_model.predict(np.array([[sensor.IR[1],
sensor.IR[2], sensor.IR[3]]]),verbose=0)
        S_result =int(np rint(S_result) *125)
        F_result =int(np rint(F_result) *300)
        if abs(S_result)>125:
            S_result=int((S_result/abs(S_result))*125)
        if abs(F_result)>300:
            F_result=int((F_result/abs(F_result))*300)

        Steering(S_result)
        Go(F_result,F_result)
        delay(100)
Close()
```

조향각과 속도를 -1,0,1로 예측 한 것을 다시 125와 300으로 올려서 사용.

최종 결과

주행 테스트 촬영 영상 별도 첨부

해당 영상 촬영 이후 알티노 좌측 센서 오류로 다른센서와 다르게 값이 매우 다르게 나와 이후 테스트가 불가능 하다고 판단하여 해당 영상을 마지막으로 제출.

학습에 있어서 애매했던 것.

사람 손으로 제어한 데이터를 사용하여 학습한 자율주행 시스템은, 전방에 아무것도 없지만 스티어링을 좌우로 컨트롤하는 데이터와 비슷한 센서값에서 뒤로 움직이거나 앞으로 움직이는 등 예측 불가능한 데이터도 같이 학습하게 됩니다. 이러한 데이터는 자율주행 시스템의 정확한 학습을 어렵게 만드는 요인으로 작용하는 것으로 판단 함.

데이터를 더 많이 쌓아서 해당 데이터들이 영향이 덜 가도록 해야한다고 판단.