Документация среды "Smart City Road"

Автор: Егор Моргунов
Магистр МГТУ им Н.Э. Баумана

Вступление

В данном документе приводится описание разработанной игровой среды для мультиагентного обучения с подкреплением. Среда имитирует движение автомобилей в условиях плотного транспортного потока с разделением агентов на кооператоров и дефекторов. В реальном мире между водителями могут возникнуть социальные напряженности в результате конкуренции за ограниченный ресурс (дорогу) и стремлении добраться до назначенной цели как можно быстрее. Моделирование сложных ситуаций на дорогах может помочь в исследовании беспилотных автомобилей и их применения в современном мире, и разработанная среда позволяет подробнее изучить взаимодействия между агентами в мультиагентных интеллектуальных системах.

Среда "Smart City Road" написана на языке Python версии 3.9, с использованием библиотек turtle, freegames и numpy. Для модификации существующих алгоритмов одноагентного и мультиагентного обучения с подкреплением, используемых для обучения среды, использовались библиотеки numpy, keras и matplotlib. В проекте представлено две различные вариации среды – одноагентная (нейросетевой алгоритм управляет лишь одним агентом) и мультиагентная (алгоритм управляет двумя агентами). Среда основана на математической модели частично-наблюдаемого марковского процесса принятия решений (POMDP), используемого как стандартизированный API в OpenAI Gym, а также на модели частично-наблюдаемых стохастических игр (POSG).

Общее описание среды

Среда "Smart City Road" представляет собой двухполосную автомобильную дорогу размером 340x340 (рис.1), по которой против часовой стрелки движутся автомобили-агенты. Агенты делятся на кооператоров (**красные**) и дефекторов (**желтые**). Агенты-кооператоры движутся по своей полосе со случайной скоростью и не могут перестраиваться. Они не получают никакой информации от среды, не имеют наград и не способны влиять на свои действия.

Агенты-дефекторы представляют собой беспилотные автомобили, которые принимают решения на основе получаемой из среды информации. Они способны двигаться с разными скоростями и перестраиваться с одной полосы на другую. Каждый агент управляется подключенной нейронной сетью, которая постепенно обучается, совершая действия и получая за них награды. Агенты должны научиться двигаться с максимальной возможной скоростью, при этом избегая столкновений с другими участниками движения.

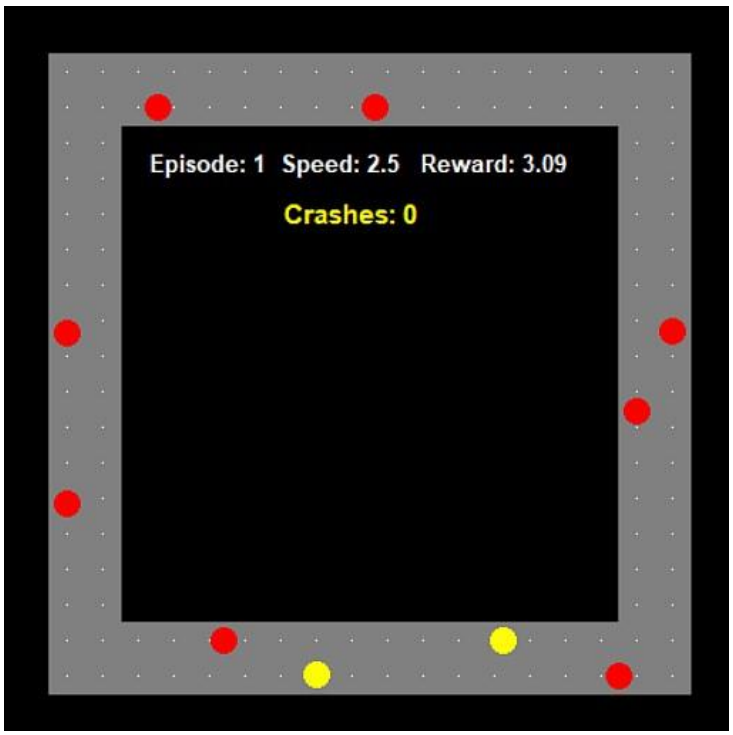

Рисунок 1

Система координат

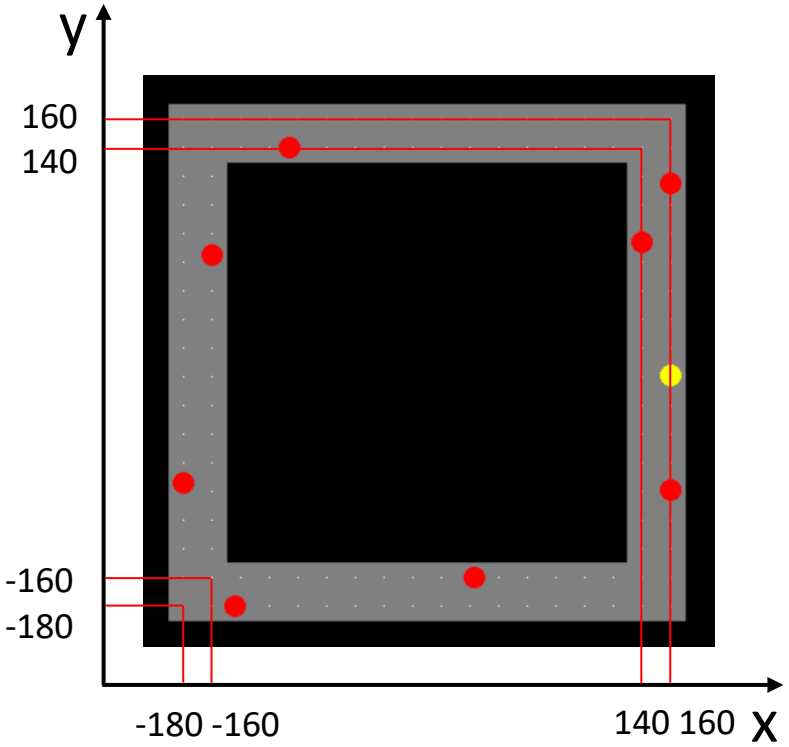

Рисунок 2

Все агенты помещаются в среду в изначально заданные координаты (рис.2), и совершают первое действие (передвижение) с заданной скоростью. Уже после скорости агенты выбирают либо случайно (кооператоры), либо обучаются выбирать самостоятельно (дефекторы). Агенты не могут пересекать полосу или выезжать за пределы установленных координат дороги – для этого их перемещение и следующие шаги контролируются с помощью функции floor(). Когда агент достигает края своей полосы, он автоматически поворачивает против часовой стрелки на 90 градусов и продолжает движение.

Агенты задаются двумя векторами – вектором координат и вектором скорости. У каждого агента есть набор возможных состояний (state space), набор возможных действий (action space) и набор наблюдений (observation space).

State Space

Набор состояний агента представляет собой список данных, которые считываются нейронной сетью каждый шаг обучения. Он содержит всю основную информацию об агенте в текущий момент времени и включает в себя:

- Местоположение агента, заданное вектором координат [x, y]
- Скорость и направление движения агента, заданные вектором скорости [x, y] Численные значения x и y позволяют определить, в какую сторону движется агент. Если, к примеру, значение x не равно нулю, то агент движется по оси x направо (если x больше нуля) или налево (если x меньше нуля) и так далее
- Набор наблюдений, состоящий из 5 численных значений

Action Space

У каждого агента-дефектора есть набор возможных действий, которые он может совершать в процессе обучения. Каждый шаг игры агенты одновременно выбирают действие и совершают его, переходя к следующему шагу. Набор действий агента включает в себя:

- Установить скорость движения равной 2
- Установить скорость движения равной 4
- Установить скорость движения равной 5
- Перестроиться – агент поворачивает в сторону соседней полосы, движется до нее, а затем поворачивается снова по ходу движения. Далее агент может как продолжить движение прямо, так и перестроиться снова, если, к примеру, впереди его движения на полосе присутствуют помехи. Агент не может двигаться прямо между полосами, для объезда он может лишь перестроиться

Observation Space

Набор наблюдений – зона видимости или угол обзора каждого из агентов в среде. Поскольку среда “Smart City Road” использует модель частично наблюдаемых стохастических игр, агенты не получают все информацию от среды, а их знания ограничены накопленным опытом и текущей зоной видимости. Наблюдения агентов вводятся для того, чтобы они научились оценивать ситуацию на дороге, избегать возможных столкновений и выбирать наиболее свободный маршрут исходя из имеющейся у них информации. Зона видимости агентов продемонстрирована на рисунке 3.

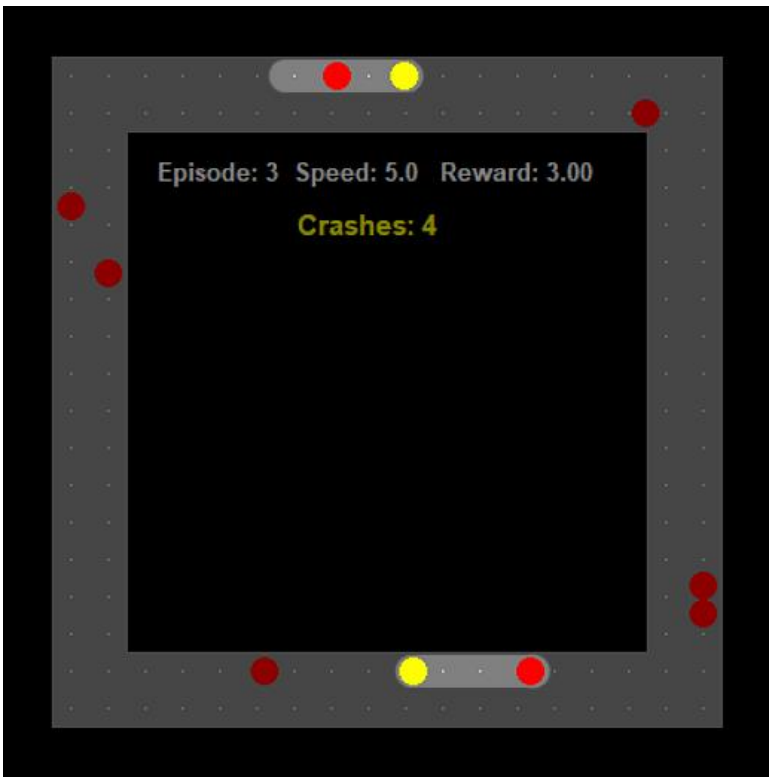

Рисунок 3

Набор наблюдений агентов представляет собой массив из 5 переменных, каждая из которых соответствует 5 координатам перед агентом во время движения. По умолчанию каждая из переменных равна нулю. В процессе игры среда сопоставляет координаты из обзора агентов с координатами всех движущихся в среде автомобилей. Если координаты одного из них совпадают с диапазоном координат зоны видимости агента, то данный автомобиль попал в поле зрения агента. После этого переменная из набора наблюдений, соответствующая диапазону координат автомобиля, становится равной единице, а среда выводит в терминал сообщение об автомобиле по курсу движения агента (рис.4).

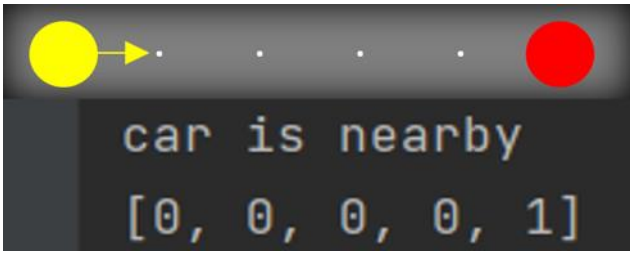

Рисунок 4

Столкновения

Для реалистичного моделирования дорожного движения была введена функция столкновения. Каждый раз, когда агент-дефектор въезжает в другой автомобиль, среда сообщает пользователю об аварии, выводя сообщение на терминал. Счетчик аварий обновляется, а агент получает штраф, который суммируется с остальными наградами. Стоит отметить, что авария фиксируется лишь в той ситуации, когда агент-дефектор врезается в другого агента, и штраф получает лишь виновник аварии.

Награды

Награды агентов формируются каждый шаг игры и напрямую зависят от совершенных действий. Награды могут быть следующие:

- Награда за движение равна скорости, с которой движется агент (2, 4 и 5)
- Награда за поворот равна половине от скорости, с которой агент перестраивался
- Награда/штраф за столкновение равна -20

Класс среды

API среды "Smart City Road" во многом основано на API Gym и включает в себя все основные функции, присущие одноагентным и мультиагентным средам. Код среды представлен в файлах "Env_Single_agent.py" и "Env_Multi_agent.py" для одноагентной и мультиагентной вариаций среды соответственно. Для каждой среды введен класс "SmartCityRoad".

Класс среды содержит следующие атрибуты и функции:

__init__(self)

step(self, action)

reset(self)

speed(self, course, n)

edge_check(self, point, course)

observing(self, agent1, agent2, agent3)

crash(self, agent1, agent2)

move(self)

Пример

Ниже представлен пример кода для тестирования среды под названием "test.py". Все действия агента выбираются случайно.

```
from Env_Multi_agent import SmartCityRoad
import random

env = SmartCityRoad()

if __name__ == '__main__':
    episode = 5
    max_steps = 100
    for e in range(episode):
        for i in range(max_steps):
            action = [random.randint(0, 4), random.randint(0, 4)]
        reward, next_state, done = env.step(action)
```