강화학습 기반의 자율주행 전기차 이동충전소

담당교수 : 홍충선 교수님
참여 연구자 : 2018110646 김연수

요 약

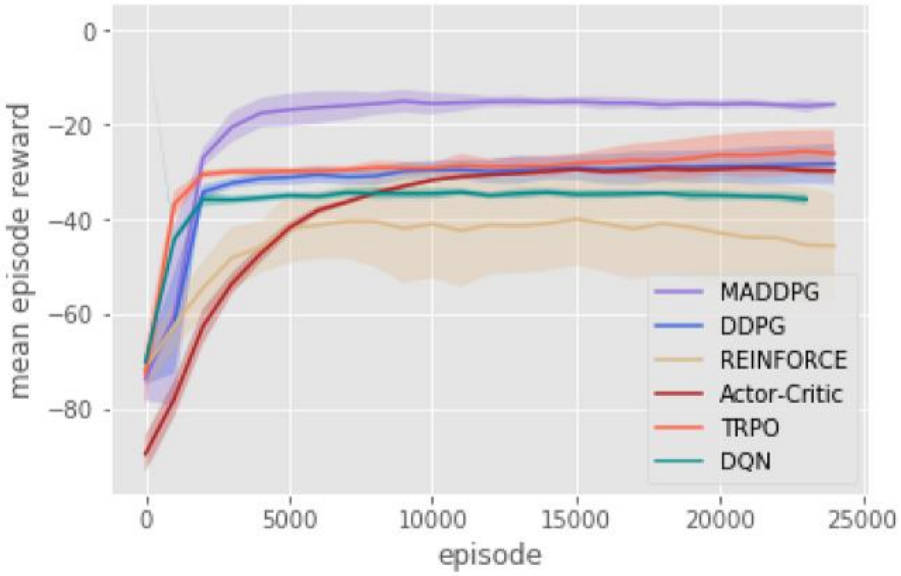
강화학습을 활용하여 실시간으로 충전을 필요로 하는 전기차의 위치를 기반으로 전기차 이동충전소가 최적의 경로로 자율주행 할 수 있는 환경을 구축한다. 이를 통해 전기차 충전소 인프라 부족 현상을 해결하고자 한다.

1. 서론

1.1. 연구배경

지난 10년간 국내에 보급된 전기차가 15만 대에 육박한 가운데 여전히 전기차 충전소 구축 속도는 이를 따라가지 못하고 있는 것으로 드러났다. 이에 따라 국내 전기차 보급에 제동이 걸릴 수 있다는 우려가 쏟아지고 있다. 특히, 지역별로 충전소 분포가 고르지 않다는 문제를 고려하여 자율주행 기반의 이동식 전기차 충전소를 제안한다. 이를 통해 전기차 충전소 인프라 부족 현상을 유연하게 대처할 수 있을 것으로 전망된다.

1.2. 연구목표



해당 프로젝트에서는 충전이 필요한 전기차 N대에 대해 이동식 전기차 충전소 M대의 최적의 주행경로를 제안한다. 이를 위해서 전기차로부터 실시간으로 전송받은 위치 정보를 바탕으로 1) 최대한 가까운 곳으로 이동하되 2) 한 번에 최대한 많은 전기차를 충전할 수 있는 위치로 이동하여 Multi Agent 모델을 학습시킨다. 또한, 충전시 경쟁상황이 발생하면 Auction Theory를 적용한다.

2. 관련연구

2.1. 최단경로탐색 알고리즘

Dijkstra 알고리즘은 최단경로 탐색 알고리즘 중 대표적이 알고리즘 중 하나이다. 이 알고리즘은 그래프 상의 출발점에서 모든 지점까지의 최단 거리를 구하는 알고리즘이다. Dijkstra 알고리즘은 간선의 가중치가 모두 양수일 때 유효하다. 모든 간선의 가중치가 음이 아닐 때, 방향 그래프 G = (V, E)에서 단일 출발점에서 최단 경로를 해결하는 알고리즘이다. 방향 그래프의 V는 정점, E는 간선을 나타내는데, 가까운 정점부터 차례로 모든 정점에 대한 간선들의 가중치를 고려하여 하나의 간선을 선택하고, 이 간선에 대한 연결된 정점을 추가하는 과정을 반복한다. 최단 경로를 찾는 방법은 가중치가 가장 가까운 정점을 선택하는 탐욕적(Greedy) 전략을 사용한다.

2.2. 강화학습 (Reinforced Learning)

강화학습은 인간이나 동물의 학습을 모방하여 작동환경에 대한 학습모형을 사용하지 않는 대표적 기계학습 방법이다. 에이전트가 어떠한 행동을 하였을 때, 좋은 피드백을 받을 경우 그 행동을 더 강화하고, 나쁜 피드백을 받을 경우 해당 행동을 하지 않도록 훈련하는 것이다. 강화학습은 액션을 수행하게 되면 환경으로부터 그에 대한 평가 피드백이 돌아오고 다음 상태로 이전하게 된다. 정책(Policy)은 특정 상태가 주어졌을 때 어떤 행동(Action)을 취할 것인지를 정해준다. 강화학습은 상태(State)와 행동(Action), 보상(Reward)의 개념을 이용하면 알고리즘을 통해 각 상황의 예측값을 결정할 수 있다. 강화학습에서 가장 중요한 요소는 보상함수를 적절히 설정하는 것이다. 보상함수를 어떻게 설정하느냐에 따라 학습의 효율이 달라진다.

2.2.1. Single Agent 강화학습

1) Q-Learning

Q-Learning은 마르코프 의사 결정 과정(MDP)을 기반으로 한 강화학습의 off-policy 기법 중의 하나이다. Q-Learning의 중요한 요소는 State(상태)와 Action(행동)이다. Q-Learning 알고리즘은 다음과 같다.

$$Q(s_t, a_t) \leftarrow (1-\alpha) \cdot \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \left(\overbrace{\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}}}^{\text{learned value}} \right)$$

위 식에서 r은 보상, s는 state로 현재의 상태를 나타내며, a는 action으로 어떤 상태에서 수행 가능한 행동을 나타낸다. 상태와 행동은 R(s,a)과 Q(s,a)로 나타낸다. 여기서 R은 보상(Reward)를 나타내고, R(s,a)은 상태에서 a라는 행동을 실행하였을 때 보상을 나타낸다. Q(s,a)는 상태 s에서 a라는 action을 실행하였을 때 Q-value값이다. A는 에이전트의 학습률을 나타낸다. $0 \le \alpha \le 1$의 범위를 가지고 학습 속도에 영향을 미치는 파라미터이다. 값의 크기에 따라 학습의 속도에 영향을 미치며, 너무 작은 값이든, 큰 값이든 값의 크기에 따라 학습이 제대로 진행되지 않을 수 있다. Q-Learning에서의 행동 선택을 위해서 생성된 Q값 중 가장 큰 값을 갖는 행동을 선택하기 위해 적용 시킬 수 있는 방법이 ε-greedy 방법이다. ε-greedy는 임의의 행동을 취하는 확률을 통해 다양한 행동집합에 의해 학습이 진행되도록 하는 선택 방법이다.

2) Policy Gradient

정책 경사 기반 강화학습은 어떤 상태에서 어떤 행동을 취할지를 결정하는 정책을 직접 구한다. 정책이란 관측 값을 입력으로 받아서 행동 값을 출력하는 함수로 매개변수 벡터 θ로 정의될 수 있다. 심층 강화학습에서는 이 함수를 심층신경망을 이용해서 근사하며, 이때 매개변수 벡터 θ는 신경망의 가중치와 편향 값이다. 정책 경사 기반 강화학습은 이 매개변수 벡터 θ를 구하기 위해서 정책 경사 기법을 이용한다. 정책 경사 기법은 θ를 구하기 위해서 경사상승법을 이용하는 방법이다. 즉, 특정 θ에 대한 목적 함수의 경사(Gradient)를 구한 후, 이 경사가 상승하는 방향으로 일정 거리만큼 θ를 업데이트하는 것을 경사가 수렴할 때까지 혹은 최대 타임 스텝만큼 반복하는 것이다.

목적 함수는 정책 πθ에 따라 행동 시 얻게 되는 누적 보상의 기대치이며, 식 (1)과 같이 표현된다.

$$J(\theta) = E_{\pi_\theta}[r(s,a)] \qquad (1)$$

목적 함수의 경사는 정책 경사 정리(Policy gradient theorem)[3]에 의해서 식 (2)와 같이 정의된다.

$$\nabla J(\theta) = E_{\pi_\theta}[Q_\pi(s,a)\nabla_\theta \ln \pi_\theta(a|s)] \qquad (2)$$

정책 경사는 목적 함수를 최대화하는 방향으로 정책의 매개변수를 식 (3)과 같이 업데이트한다.

$$\theta \leftarrow \theta + \alpha\nabla J(\theta) \qquad (3)$$

2.2.2. Multi Agent 강화학습

멀티 에이전트 강화학습(MARL: Multi-Agent Reinforcement Learning)이란 주어진 환경에서 두 개 이상의 에이전트가 협업 또는 경쟁을 통해 높은 보상을 얻을 수 있는 행동 정책(Policy)을 학습하는 기술로 정의할 수 있다. 기존 싱글 에이전트 중심의 강화학습 기술에서 고려하고 있는 탐색-이용 딜레마(Explore-exploit dilemma), 부분 관측 가능성(Partial observability)에 따른 문제들뿐만 아니라 멀티 에이전트 환경이 갖는 고유의 비정상성 특성과 에이전트 간의 신뢰할당(Credit-assignment) 문제까지 추가로 고려해야 한다.

2.3 Auction Theory

/* Bidding Submission of d_i */
Each buyer d_i requests the energy trading and provides its bid vector, $\mathbb{B}_i$
/* Winning Bid Determination at s_j */
$x_{ij}^{temp} = 0, p_j^t = 0, \forall d_i \in \mathcal{D}; s_j \in \mathcal{S}; W_j^{can} = \emptyset.$
Determine the set of feasible buyers $W_j^{temp} = \{d_i | e_i \leq E_j, t_i \leq T_j, \forall d_i \in \mathcal{D}\}$.
Sort the bids of buyers $d_i \in W_j^{temp}$ in non-increasing order:
$W_j^{order} = (d_{j_1}, d_{j_2}, ..., d_{j_K})$ such that $b_{j_1 j} \geq b_{j_2 j} \geq ... \geq b_{j_K j}$ with $K = |W_j^{temp}|$.
Pick out $|c_j|$ bidders having the highest bids

$$W_j^{cons} = \begin{cases} (d_{j_1}, d_{j_2}, ..., d_{j_{c_j}}), & \text{if } c_j < K, \\ W_j^{order}, & \text{otherwise.} \end{cases}$$

if $\sum_{d_i \in W_j^{cons}} e_i \leqslant E_j$ **then**
  $W_j^{can} = W_j^{cons}$; $x_{ij}^{temp} = 1, \forall d_i \in W_j^{can}$.
  $p_j^t = b_{mj}$ where

$$m = \begin{cases} \arg\max_i \{d_i | d_i \in W_j^{order} \setminus W_j^{cons}\} & \text{if } c_j < K, \\ \arg\min_i \{d_i | d_i \in W_j^{order}\} & \text{otherwise.} \end{cases}$$

if $\sum_{d_i \in W_j^{cons}} e_i > E_j$ **then**
  $h = \arg\max_h \sum_{h'=1}^{h} e_{h'} \leq E_j, \forall d_{h'} \in W_j^{cons}$.
  $W_j^{can} = \{d'_h | 1 \leq h' \leq h\}; x_{h'j}^{temp} = 1$.
  $p_j^t = b_{(h+1)j}$.
/* Final seller determination at d_i */
for $j = 1$ *to* S **do**
  $x_{ij} = 0; p_i^d = 0$.
if $\sum_{j=1}^{M} x_{ij}^{temp} = 0$ **then**
  $x_{ij} = 0, \forall j; p_i^d = 0$.
if $\sum_{j=1}^{M} x_{ij}^{temp} = 1$ **then**
  for $j = 1$ *to* M **do**
    if $x_{ij}^{temp} = 1$ **then**
      $x_{ij} = 1; p_i^d = p_j^t$.
if $\sum_{j=1}^{M} x_{ij}^{temp} > 1$ **then**
  for $j = 1$ *to* M do **do**
    if $x_{ij}^{temp} = 1$ **then**
      $U_{ij}^d = (v_{ij} - p_j^t)e_i$.
  $j^* = \arg\max_j \{U_{ij}^d | x_{ij}^{temp} = 1, \forall j \in M\}$.
  $x_{ij^*} = 1; p_i^d = p_{j^*}^t$.

경쟁상황이 발생했을 때, bidder는 입찰가를 비공개로 제시하고, auctioneer 입찰가 중 최고가를 winner로 결정한다. 이때, multi agent 상황에서는 auctioneer끼리 서로의 상황을 알 수 없으며, 해당 알고리즘에서는 bidder인 EV(Electronic Vehicle)이 필요로 하는 energy list 중 최고가의 입찰가를 순서로, auctioneer인 MCS(Mobile Charging Station)이 제공할 수 있는 energy 까지만 제공한다.

3. 프로젝트 내용

3.1. 시나리오

1. **전기차 충전소가 action space를 기반으로 상태를 update한다.**
   이때, 이동은 상,하,좌,우,no operation 총 5가지가 가능하다.

2. **전기차는 근처의 충전소가 어디에 위치해 있는지 정보를 바탕으로 가까운 충전소로 이동한다.**
   - 이때, 이동은 상,하,좌,우,no operation 총 5가지가 가능하다.
   - 만약, 하나의 전기차 주변에 두 대 이상의 전기차 충전소가 존재하는 경쟁상황이 발생한다면 auction theory를 활용한다. 더 적은 가격으로 충전을 할 수 있는 충전소에서 충전한다.

3. **step을 진행하면서 1,2번 과정을 반복한다.**

3.2. 요구사항

1. 빠른 시간내에 전기차 충전소가 전기차를 충전한다.
2. 경쟁 상황이 발생했을 때, 전기차가 더 저렴한 가격으로 전기차 충전소에서 충전한다.

4. 프로젝트 구현단계

4.1. 정의

1. Action Space

   주어진 환경에서 가능한 모든 action의 set을 의미한다. 해당 프로젝트에서 action space는 전기차 또는 전기차 충전소가 이동할 수 있는 이동 방향으로, 상,하,좌,우와 이동하지 않음, 총 5가지 이다.

2. Observation Space

   Environment의 현재 상태에 대한 정보를 의미한다. 이때, agent인 전기차 충전소는 자신의 근처에 있는 전기차를 알 수 있다. 하지만 모든 전기차의 위치는 알 수 없다.

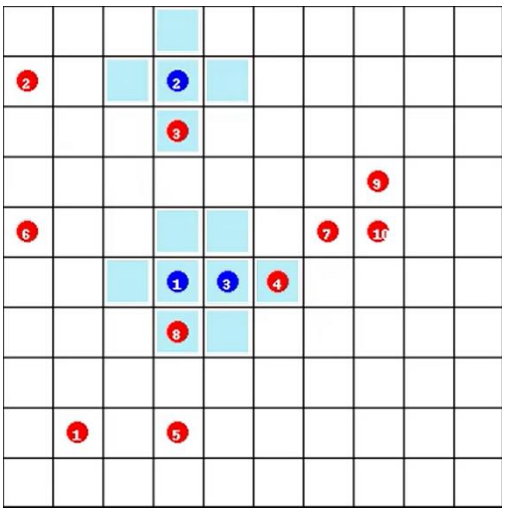
3. Policy

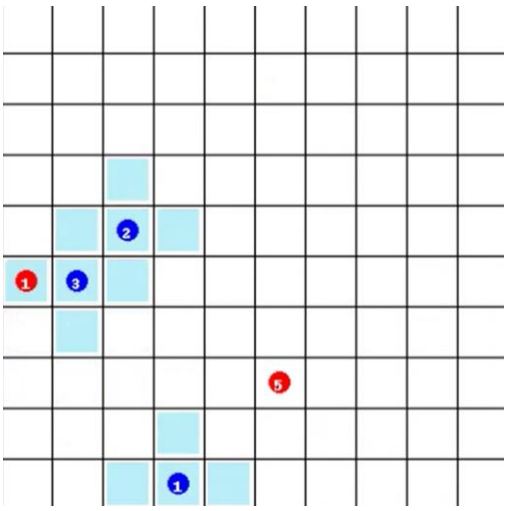
   Agent가 어떤 Action을 취할지 선택하는 rule이다. 확정적(Deterministic)일 수도 있고, 확률적(Stochastic)일 수도 있다. 해당 프로젝트에서는 Stochastic한 방법으로 접근했고, 전기차가 이동하는 방향의 각각의 확률을 (0.175, 0.175, 0.175, 0.175, 0.3)으로 정의했다.

4.2. 환경

1. 일반적으로 주행시 임의의 방향으로 이동할 수 없고, 주어진 map에서 주어진 차선을 지켜야 함을 고려하여 grid한 환경속에서 움직이는 것이 좋다고 판단하여 OpenAi의 gym 오픈소스 중 gridworld 예제를 변형하여 환경을 구축하였다.
2. 전기차 n대와 전기차 충전소 m대의 문제상황을 구현하기 위해 multi agent 환경을 구축하였다.

5. 프로젝트 결과





다음 그림에서 빨간색은 전기차이고 파란색은 전기차 충전소이다. 8대의 전기차와 3대의 전기차 충전소로 테스트 했다. 둘다 시간이 흐름에 따라 이동하며, 전기차는 전기차 충전소와 가까운 방향으로 이동하며 전기차 충전소는 상,하,좌,우에 있는 전기차 충전소를 충전한다. 충전이 끝난 전기차는 화면에서 사라진다. 테스트 후반부로 갈수록 전기차가 충전됨을 볼 수 있다.

6. 프로젝트 기대효과

기존의 전기차 충전은 충전소 또는 충전 주차장에서만 가능했다. 이는 시간, 공간적 제약이 컸고, 많은 사람이 사용하기에는 비효율적이라는 판단이 된다. 따라서 해당 프로젝트를 통해 전기차 이동충전소가 자율주행을 하며 필요로 하는 운전자를 찾아가는 시스템을 개발하면 충전 인프라 부족현상을 해소할 수 있을 것으로 예상된다.

7. 추후 연구 계획

모델을 더 경량화시켜 전기차와 전기차 충전소를 임베디드 환경으로 구축하여 모델이 동작하게 한다.

8. 참고 문헌

[1] 구다솔, 이태경, 강화학습 기법을 이용한 최적경로탐색, 2014

[2] Ryan Lowe, Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments, 2020

[3] OANH TRAN THI KIM, Distributed Auction-Based Incentive Mechanism for Energy Trading between Electric Vehicles and Mobile Charging Stations, 2022