Ride or Die: Optimizing Electric Vehicle Charging, Discharging or RoboTaxi using Q-learning

CS221 Project: Final Report

Team: Neural Energy

CA Mentor: Apoorva Dixit (dixit5@)

Hannah Hagen	(hlhagen@)	Henry Daniels-Koch	(hdaniels@)
Marcelo Fernandez	(marcefer@)	Rohit Makhija	(rohitm@)

Introduction: Problem Overview

We explore the problem of optimizing the use of an Autonomous Electric Vehicle while it is not in use by it's owner, choosing between using the vehicle as a robotaxi vs. discharging the battery to put power back on the grid or instead charging the vehicle's battery. We use real electricity prices for charging, an incentive rate for returning power to the grid, and handcrafted rider demand data. We model this problem using Markov Decision Processes and Reinforcement Learning, with a single vehicle.

Literature Review

The 2018 case study Modeling shared autonomous electric vehicles: Potential for transport and power grid integration[2] explores a linear programming methodology to simulate deployment of a fleet of shared autonomous electric vehicles in Tokyo, concluding that such efforts could replace 7 to 10 private cars while also operating as operating reserve to the power grid.

- Their work uses linear programming. We plan to use MDPs and Q-learning.
- They simulated the conditions for a fleet of shared vehicles deployed in a city, replacing private cars. We are exploring one EV in isolation and how we might maximize reward for the EV owner.
- Their case study represents a steady-state where a city government or private operator invests in a fleet of shared vehicles. We are exploring a more nascent market, where individual owners of autonomous electric vehicles might be incentivised to generate earnings from privately owned autonomous vehicles while they are otherwise not in use.

Dataset

Parameters were hand selected from a combination of real and simulated data to model changes in a battery's state of charge, and the monetary rewards of different actions.

Charging transitions:

Action	Energy (kWh/30min)	
Discharge	-5	A 30min discharge releases 5kWh of energy[1]
Charge	+125	Charge at 250kWh/hour, inspired by Tesla supercharing rates[4]
Ride	-7.5	A typical 30min ride uses 7.5 kWh of energy[3]

Pricing:

- Charging prices are based on real electricity price schedules, and are higher during peak (4pm-10pm), and extreme peak hours (7:30pm-9pm).
- Discharging rewards are a function of charging rates. They can be derived using a multiplier applied to the corresponding charging rate.
- Ride prices are modeled as a distribution, with higher ride prices during peak morning and evening traffic hours. Each episode is sampled from this distribution to model the variability/uncertainty in our system.
    - The agent suffers a penalty of $10 if it tries to ride without sufficient state of charge to complete the ride.

Charging Rates, USD per kWH per 30min				
		Avg Charging Rate	Charging Variance	Discharge Multiplier
0000 - 1600	Off Peak	$0.24	0.02	$\sim U(0.8, 1)$
1600 - 1930	Peak	$0.48	0.05	$\sim U(2, 4)$
1930 - 2100	Extreme Peak	$1	0.01	$\sim U(5, 20)$
2100 - 0000	Off Peak	$0.24	0.02	$\sim U(0.8, 1)$

$X \sim U(0.8, 1.0)$:= X takes values in a Uniform distribution between 0.8 and 1.0

Average Ride Rewards, USD per 30min		
0000 - 0600	Night	∼ Poisson($1)
0600 - 0700	Off Peak Daytime	∼ Poisson($5)
0700 - 0900	Peak Daytime	∼ Poisson($15)
0900 - 1700	Off Peak Daytime	∼ Poisson($5)
1700 - 1900	Peak Daytime	∼ Poisson($15)
1900 - 2200	Off Peak Daytime	∼ Poisson($5)
2200 - 0000	Night	∼ Poisson($1)

Figure 1 show synthetic electricity and ride price data for two different episodes. At each time step, electricity and ride prices are sampled from distributions described above, resulting in the variability seen between episodes in Figure 1.

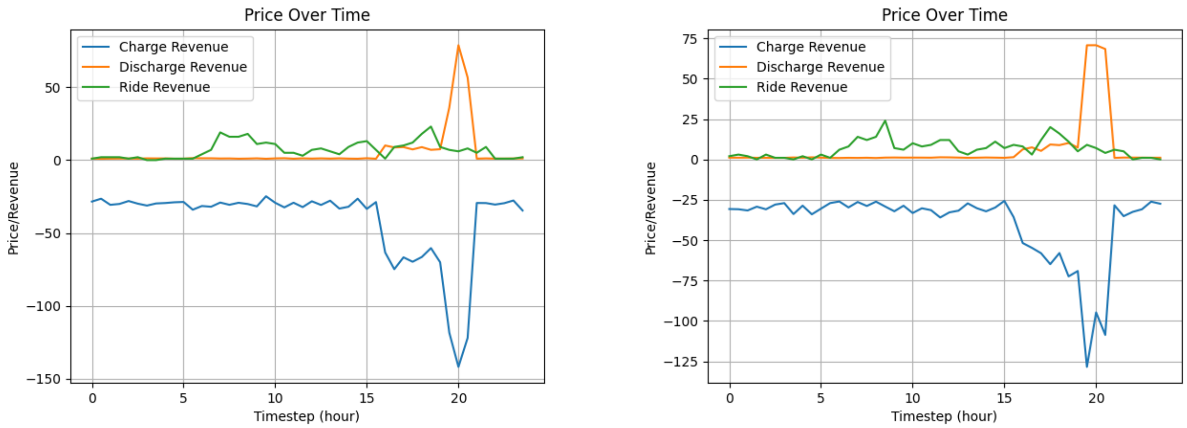


Figure 1: Prices simulated for 2 different episodes

Main approach

We treat the state of charge of the EV as its current state 's'. Now considering the actions it can take, the rewards or costs it might incur, and the resulting change in its state of charge, we find that a Markov Decision Process (MDP), suitably models our problem with the following components:

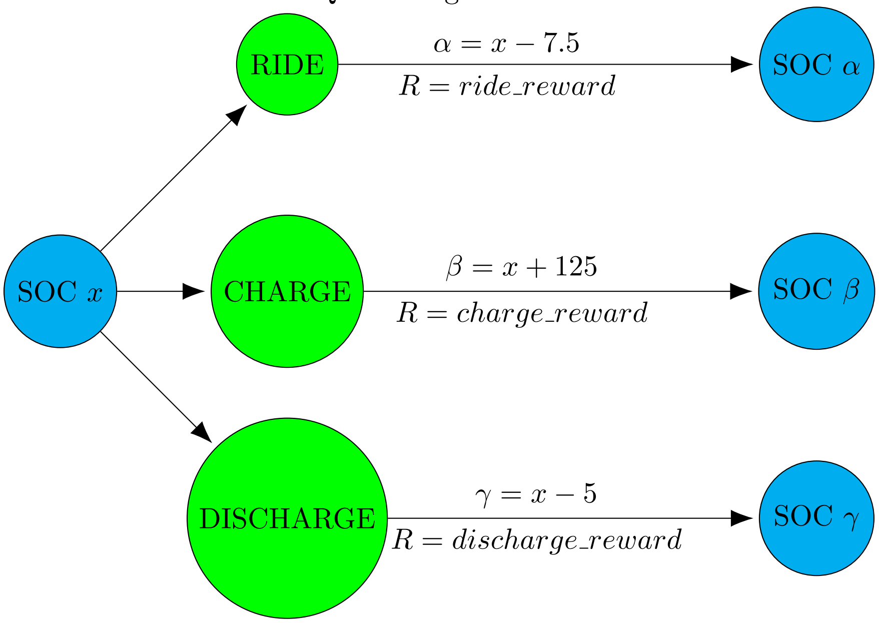
1. **State (S)**:
    - Current State of Charge (SOC) of the EV battery.
    - Price for discharging electricity back to the grid in the previous timestep.
    - Cost of purchasing electricity from the grid in the previous timestep.
    - Ride price in the previous timestep
    - Current Timestep (in 30 minute increments)
2. **Actions (A)**: The decisions the EV owner can make.
    - Charge the EV.
    - Discharge the EV to the grid.
    - Provide a ride.
3. **Rewards (R)**: The immediate payoffs received after taking an action.
    - Money spent on charging the EV.
    - Money made from discharging to the grid.
    - Money earned from providing rides.
    - Penalties for actions like overcharging or depleting the battery excessively.
4. **Transitions (T)**: These describe how the state changes as a result of the actions taken.
    - The SOC changes based on charging, discharging, or providing a ride.

- Time progresses, influencing electricity prices and ride demand.
- Electricity prices are deterministic (based on a fixed pricing schedule).
- Ride prices, however, are uncertain. They are likely based on factors such as time of day, day of week, traffic, weather, etc.

Our goal is to find the optimal policy that maximizes rewards, in terms of payout to the EV owner. Our transition probabilities are unknown due to uncertain ride prices. So we use reinforcement learning to learn the optimal policy. We aim to compare a few reinforcement learning algorithms:

- Q-learning with linear regression
- Deep Q-learning

We did not use Tabular Q-Learning because our state size was too large.



Legend:

- States represent the State of Charge of the Electric Vehicle
- Chance nodes represent the actions taken

Evaluation Metric

We compared our baseline policy against the policies learned from each of our reinforcement learning approaches using the following metrics:

- Average Total Reward over 100 episodes (Primary metric)
- Rate of convergence, i.e. number of episodes until convergence (Secondary metric)

Baseline

Our baseline policy chooses the charge/discharge/drive action based on simple rules. See the Results and Analysis section for more details

$$\pi_{\text{baseline}} = \begin{cases} \text{charge} & \text{if battery level is less than 10\%} \\ \text{discharge} & \text{if discharge revenue is greater than ride revenue in previous timestep} \\ \text{give ride} & \text{if ride revenue is greater than discharge revenue in previous timestep} \end{cases}$$

Results and Analysis

	Baseline	Q-Learning w/ Linear Reg	Deep Q-Learning
Avg Total Rewards (avg over 100 episodes)	\$251	\$190	\$231

Note: avg total rewards was prone to high variance for our q-learning models when only averaged over 100 episodes. Due to run-time constraints, we restrict our test length to 100 episodes, but given more time, we'd extend the test length.

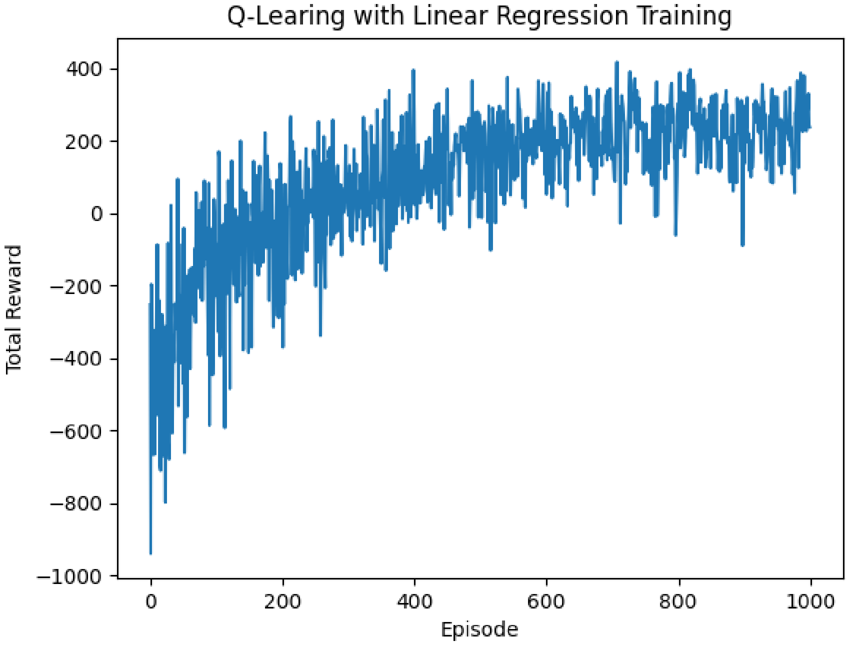


Figure 2: Q-learning Training: total rewards per iteration/episode

Our baseline policy is currently outperforming both of our q-learning implementations (with linear regression and with a deep neural network). Deep Q-learning is performing better than Q-learning with linear regression. We believe this is because of the non-linear relationship between our state variables (state of charge, time) and the expected reward due to price having a non-linear relationship with time (see Figure 1). As a result, deep Q-learning which is capable of learning non-linear relationships outperforms Q-learning with linear regression.

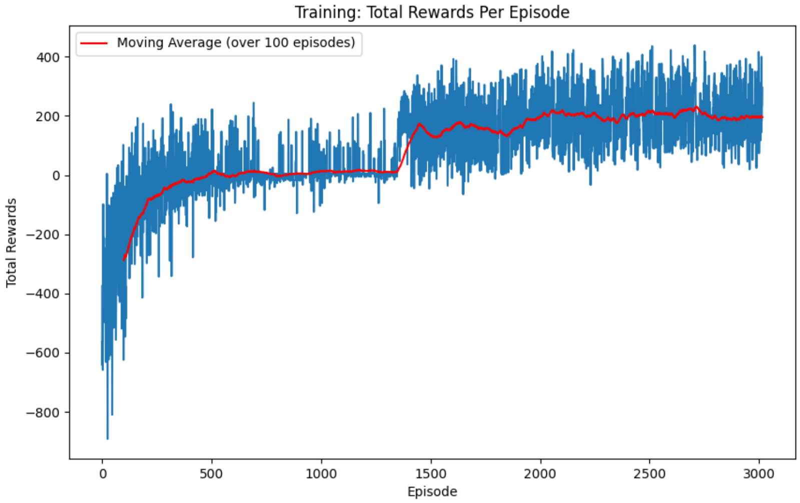


Figure 3: Deep Q-Learning Training: Total rewards per iteration/episode

We implemented a Deep Q-Learning method using a simple multi-layer perceptron. Training over 3000 episodes, we observe the following. The method starts by obtaining rewards around -300 on average with high variability. It then learns fairly rapidly how to avoid negative rewards from between episode 500 to 1400. When we look at the policy, we see that the action is to discharge at every time step in order to avoid the high charging cost of the charge action and the penalty associated with giving a ride and not having enough charge. However, this policy is myopic when evaluated over many timesteps. We would hope the model would learn to accept the high charge costs during early timesteps and prepare for the peak discharge prices later in the day (between 4-9pm). These are the times it can early significant rewards. At timestep 1400, we observe a ”breakthrough” moment for the agent, in that it discovers this long term planning oriented policy where it can be beneficial to charge. After episode 1500, we observe small incremental improvements and fluctuations, but no serious improvements. When comparing the baseline average reward of $250to190$, we observe the baseline performing much better.

Error Analysis

We see some suboptimal or erroneous behaviors

- We would expect the EV to prioritize charging at night, then prioritize discharging when electricity pricing is at extreme peak and chose between riding and discharging at other times
- In practice, after several episodes, our implementation has not learned to take full advantage of charging prior to the 8pm peak in discharge price. While it has learned to discharge during

the peak discharge price, the battery level is often already low and thus the amount of money earned in discharging to the grid is limited and it leaves money on the table.

- We suspect our Deep Q-learning model got stuck in a local optimum.

Code

Github Repository

Future Work

Given more time:

- We would try different episode lengths, likely longer episode lengths
- We would explore if increasing or reducing time intervals (30min time steps = 48 time steps over 24 hours, vs 2hr time steps = 12 time steps over 24 hours) to explore if the rate of convergence improves.
- We modelled ridership demand with handpicked data, but this could be modelled using real ridership data.
- We would like explore modeling multiple vehicles at once, and eploring larger simulations like the one in Modeling shared autonomous electric vehicles: Potential for transport and power grid integration[2].
- We could like to explore other modeling techniques, like applying CSPs and Bayesian nets to this problem.

References

[1] Scott Evans. “How the Ford F-150 Lightning Can Power Your Whole House”. In: *Motor Trend* (). DOI: `https://www.motortrend.com/features/2022-ford-f-150-lightning-home-power/#:~:text=Ford%27s%20gone%20a%20step%20further.%20Not%20only%20does%20the%20Lightning%20include%20the%202.4%2DkW%20system%20and%20eight%20outlets%2C%20but%20an%20optional%209.6%2DkW%20system`.

[2] Riccardo Iacobucci, Benjamin McLellan, and Tetsuo Tezuka. “Modeling shared autonomous electric vehicles: Potential for transport and power grid integration”. In: *Science Direct* (). DOI: `https://www.sciencedirect.com/science/article/pii/S0360544218310776?via%3Dihub#sec5`.

[3] Frank Markus. “What Happens When Your Ford F-150 Lightning EV Battery Runs Out of Range?” In: *Motor Trend* (). DOI: `https://www.motortrend.com/reviews/2023-ford-f-150-lightning-xlt-yearlong-review-update-8/`.

[4] *Tesla Supercharger*. Accessed: 2024-06-03. DOI: `https://www.tesla.com/supercharger`.