Deep Reinforcement Learning in Simulated Automated Parking

Lear Du, Hui Xu
Stanford University

Introduction

Automated car parking is a necessary feature for self-driving cars that have recently enjoyed fast progress and wide attention. In this project, we attempt to use reinforcement learning models to simulate automated car parking problem. Our objective is the following:

- Develop model using reinforcement learning to automatically park a car adhering to 2 wheel kinematics
- Train the model so the car can park from any feasible parking configuration

Since we have a continuous state space, we decided to use Q-learning with function approximation and tried two types of function approximators: a linear model, and a neural network based model.

Model

We frame the task of automated parking as a deterministic MDP.

- **State:**
  1. Relative position between car and parking lot $(x_c - x_p, y_c - y_p)$
  2. Orientations of car and parking lot θ_c, θ_p
  3. Speed of car v_c, Steering angle ϕ_c
  4. Number of steps taken

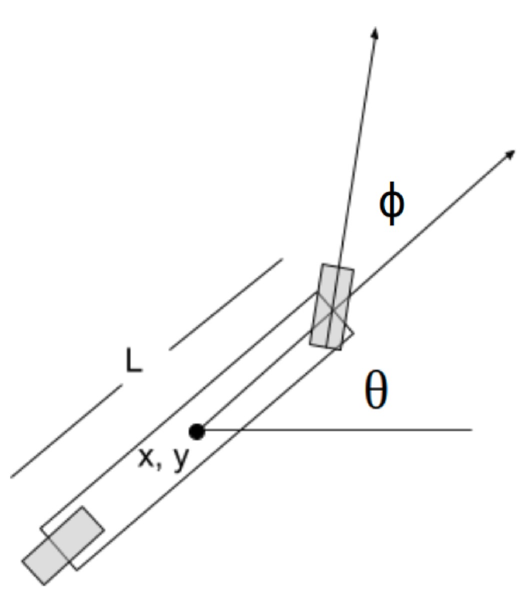


Fig. 1: kinematics

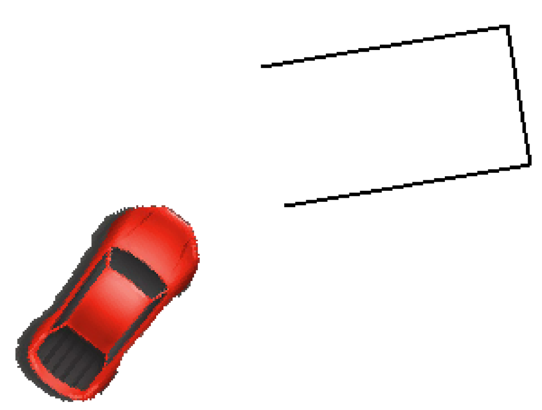


Fig. 2: simulation

- **Action:** Choose from the six discrete actions{Left Straight, Right Straight, Straight, Left Reverse, Right Reverse, Back Reverse}
- **Transition:** Two-wheel kinematics updated per 0.1 second

$$dx_c = v_c \cos(\theta_c) dt$$
$$dy_c = v_c \sin(\theta_c) dt$$
$$d\theta_c = v_c \tan(\theta_c) dt / L_c$$

- **Reward:**
  - +1 if the car moves towards the parking center
  - -100 if the car moves away from parking center
  - +100 - #steps if the car is within position threshold of the parking lot
  - +250 - #steps if the car is within position and orientation threshold of the parking lot

Oracle

Our oracle is to create a manual mode where humans can use the keyboard to park the car.

Architecture

1. **Baseline linear approximation**: We approximate $Q(s,a)$ by a linear function $Q(s,a) = w \cdot \phi(s,a)$, where features are defined as follows:
$$\phi(s,a) = \left(v_c, \theta_c, x_c - x_p, y_c - y_p, \phi_c, \arctan 2\left(\frac{x_c - x_p}{y_c - y_p}\right) - \theta_p \right),$$
where the last feature defines the delta angle between the heading from car center to parking center and the heading of the car.
2. **Neural network**

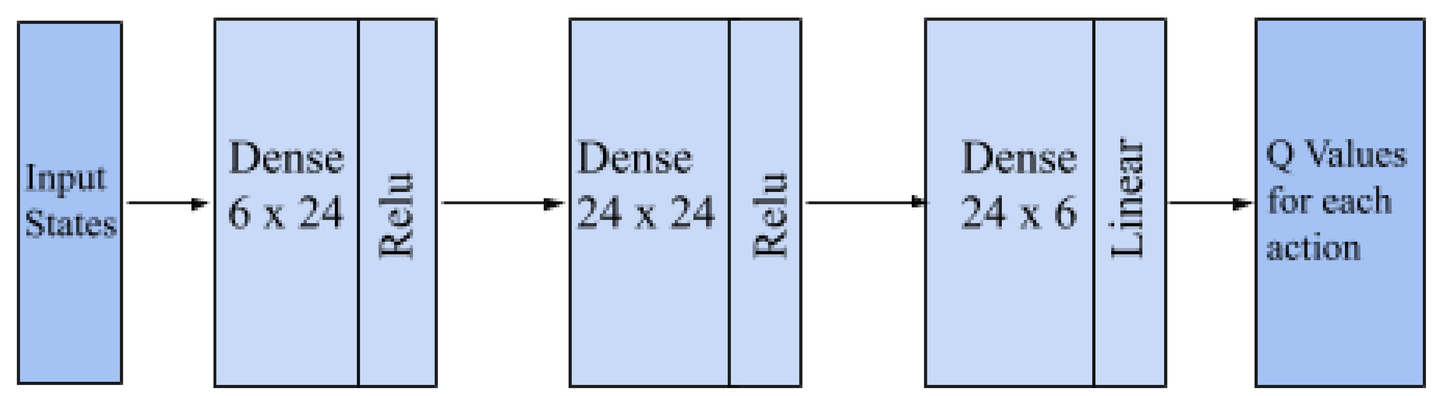
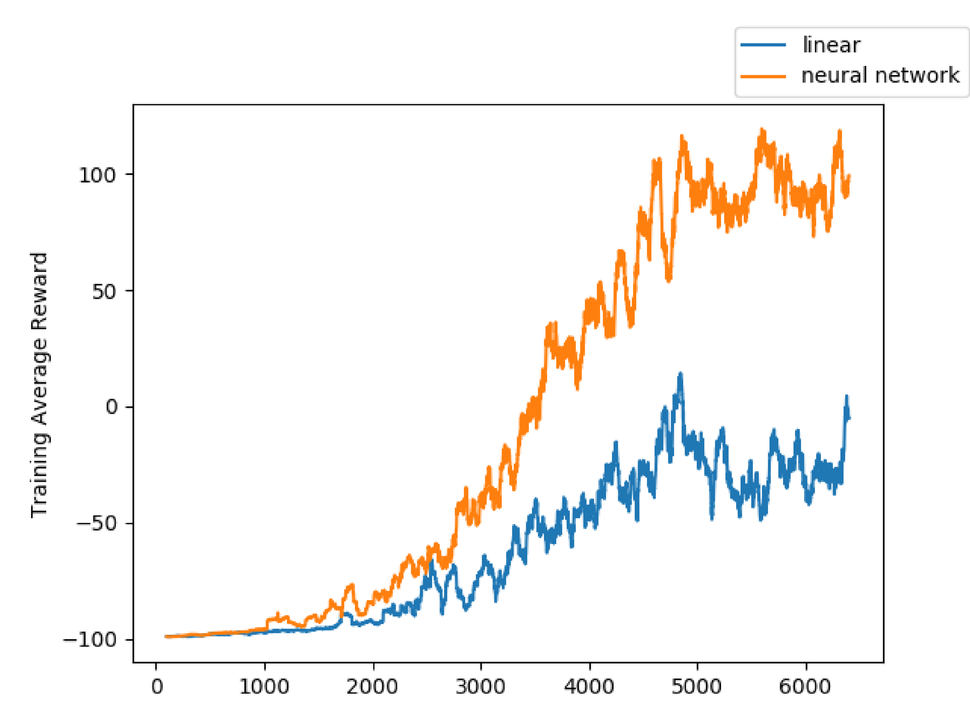


Fig. 3: neural network architecture for deterministic action rewards

- two hidden dense layers with RELU activation, one output layer with linear activation
- predict the Q value for each state-action pair and optimal action as $\texttt{argmax}_a Q(s,a)$
- use experience replay
  - At every iteration we initialize the car at a random position and orientation. Store the state action reward in the experience replay memory.
  - At the end of each iteration, take a sample batch of replay instances from memory and run fitting

Training

For training, linear model peaked at score of 14.18, while neural network model peaked at 118.59.



Evaluation and comparison:

- For linear baseline and the neural net, the model was saved every 200 iterations. For each saved model we ran 50 trials (each time initializing car at a random location that is feasible to park from) and calculated the average reward and the parking success rate (satisfy both distance and orientation threshold).
- For the human evaluation, we performed 50 trials where human used key controls to try to park car.

The following table summarizes the result of the three models:

	Linear	Neural Net	Human
Best Avg Reward Score	38.4	113.1	124.7
Best Parking Success Rate	18%	60%	86%

For the linear and neural network models we fixed the initial car to parking lot distance ($d = 4.0625, 8.125, 12.1875$) and adjusted the initial correction angle $\theta_p - \theta_c$ from 0 to 30 degrees and analyzed the parking success rate. Overall the neural network performed better than the linear model. Each saw a success rate drop when the correction angle was above a certain threshold. The linear model was not able to park at all when the correction angle was too high.

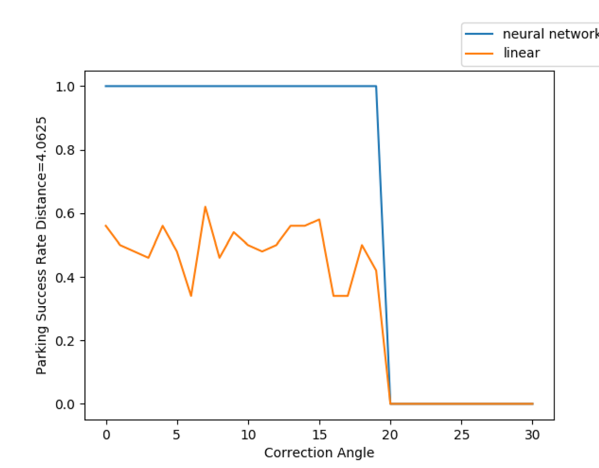


Fig. 5: d=4.0625

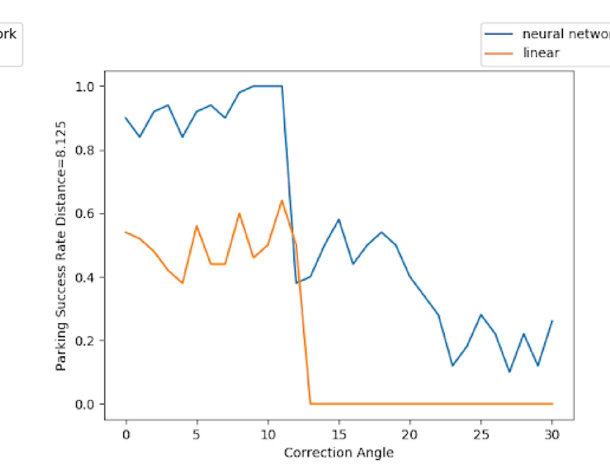


Fig. 6: d=8.125

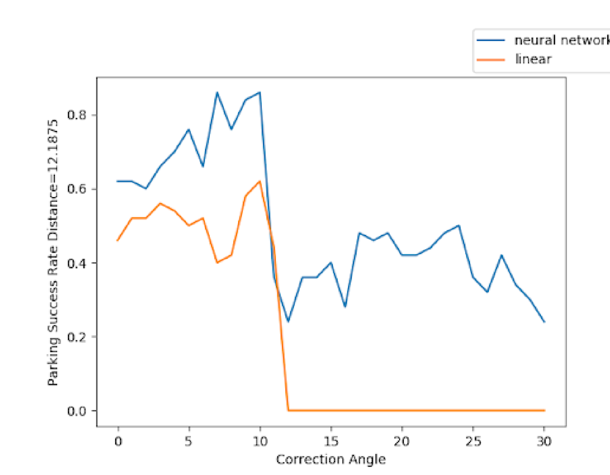


Fig. 7: d=12.1875

Conclusion and Future Work

Neural network model performed about the same as a human oracle when the correction angle required was small enough. For high initial correction angles, the car was not able to turn sharp enough and missed the parking spot. Future work could be conducted to perform more difficult parking maneuvers such as curb pullover or parallel parking.