Reinforcement Learning: Motivation and Common Questions

Earl Wong

Learning

Supervised Learning

Unsupervised Learning

Reinforcement Learning

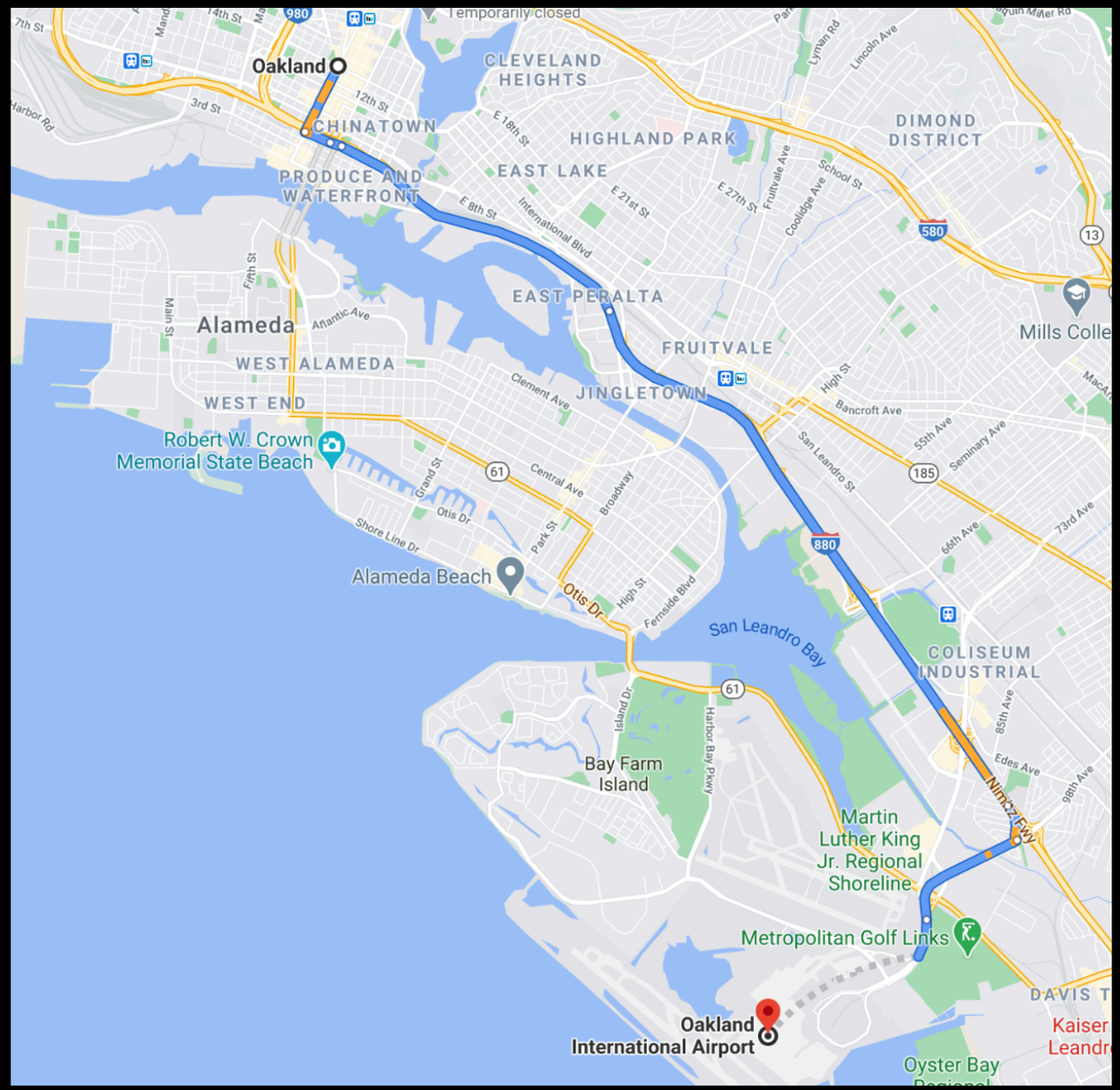
Motivation

- Decisions, Decisions, Decisions
- Everyday, you make decisions.
- These decisions result in actions …

Reinforcement Learning

- … and these actions have consequences (rewards)…
- … and these consequences (rewards) are often delayed.
- Objective: You want to make the best possible (optimal) decisions / **take the best possible sequence of actions**, to maximize or minimize an end objective / goal (total expected return).

Example:
Travel From Oakland, CA to Manhattan, NY

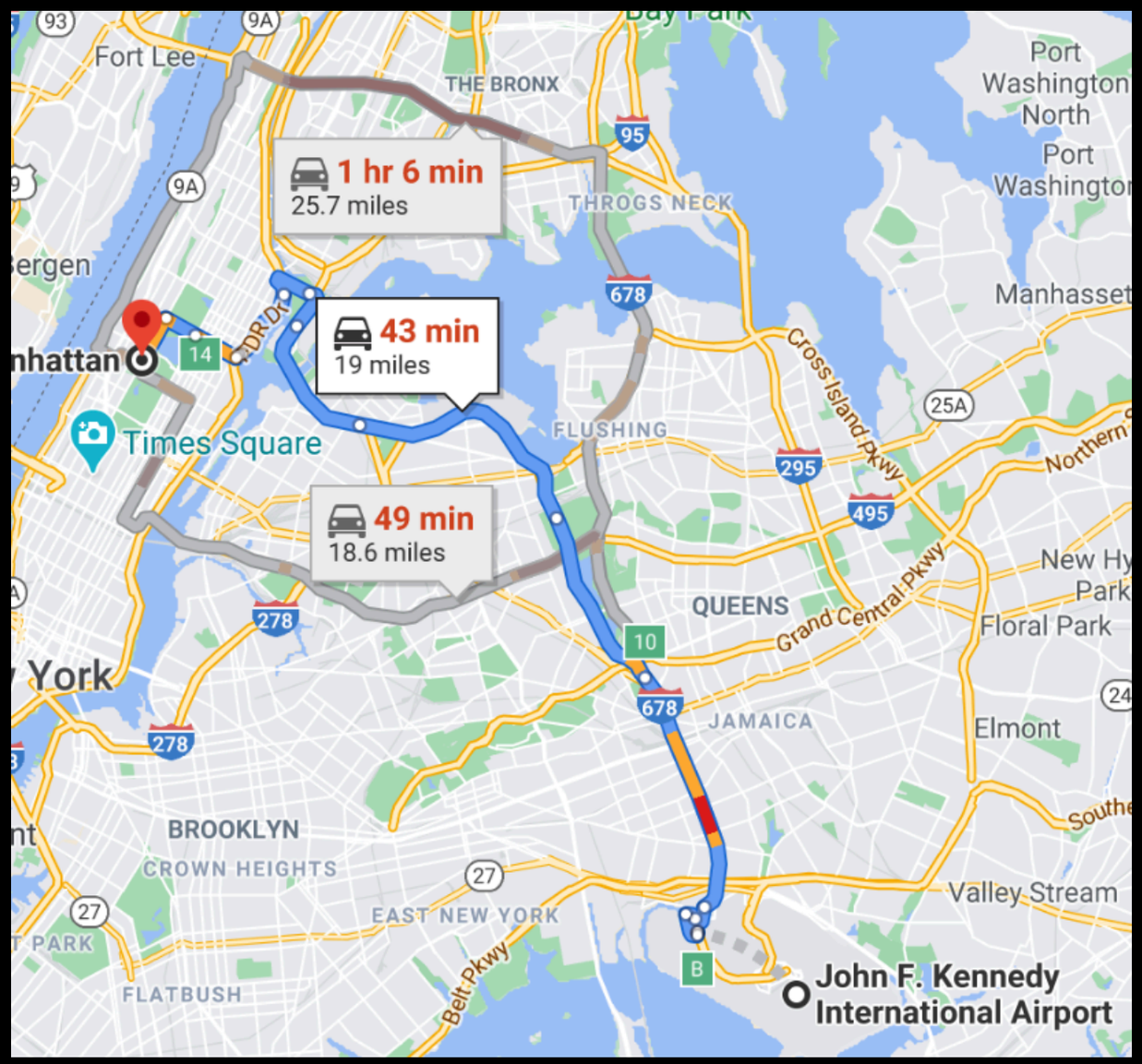


Should I 1) take BART to the airport, 2) take an Uber / Lyft to the airport or 3) drive to, and park at, the airport?

Example:
Travel From Oakland, CA to Manhattan, NY

Should I 1) take a direct flight from Oakland International Airport to JFK Airport,
2) have a layover in San Antonio, or 3) have multiple layovers at several airports?

Example:
Travel From Oakland, CA to Manhattan, NY



Should I travel from JFK Airport to Downtown Manhattan by 1) AirTran, 2) Shuttle or 3) Uber / Lyft?

If Uber / Lyft, which route should the driver take?

Objective / Goal

- What are you trying to maximize (minimize)?
- Time
- Cost
- Inconvenience / Stress

Model vs Model Free

- If the traveler had complete knowledge of all aspects of his / her travel options, the traveler is operating in the model based scenario.

- In contrast, if the traveler knew absolutely nothing about his / her travel options, the traveler is operating in the model free scenario.

- In the model free scenario, the traveler needs to thoroughly explore the environment, to determine his / her various options.

- He / she might “hit upon” all of the options previously described in the model based scenario, and also, discover new options.

- For example: He / she might determine that driving from Oakland to Manhattan is the best option.

Model Based Subtleties

- In some situations, the model is well defined, and is completely described.
- This is true in situations involving games (chess, checkers, pong, donkey kong, etc.), since the rules of the game are completely known.
- In other situations, the model exists, but the parameters of the model need to be learned.
- In our example, the 3 legs of the trip could be defined. However, the options for each leg and / or the details associated with each option (such as time or cost) may need to be learned.

What is True

- In a model free environment, the best actions need to be learned through trial and error.
- In a model based environment with known model parameters, the best actions need to be learned, given the rigid constraints.
- In a model based environment with unknown model parameters, both the model parameters and the best actions need to be learned.

What is this “Agent”?

- An agent is the entity that lives in, and interacts with the environment.
- The agent functions in the environment, by taking different actions in the environment.
- The goal of the agent is to perform actions that maximize (minimize) expected return.

What is this “Agent”?

- How does an agent accomplish this?
- The agent must take “optimal” actions, for different state(s) / observation(s).
- In RL, these actions are described by a policy.
- In a deterministic environment, the policy says: “perform action XYZ” to maximize (minimize) the end objective / goal.
- In a stochastic environment, the policy may say: “perform action ABC 20% of the time and action DEF 80% of the time”, for a given scenario / state.

Goal

- Of Agent: Take actions to maximize (minimize) expected return.
- Of RL: Determine a policy / policies to allow the agent to achieve this objective.

Action Space of Agent

- The agent can operate in an environment requiring discrete or continuous actions.

- For example, balancing a pole on a car (move left or right) or playing PacMan (move left, right, up, down) are examples of discrete action spaces.

- In contrast, the motion of robotic creatures (adjust the torque of joint xyz to some value between [-1, 1] is an example of a continuous action space.

Policy of Agent

- But, how is the policy determined?
- The policy needs to be learned.
- In a model free environment, the policy is learned by trial and error.
- At the other extreme, if the complete model of the environment is known (such as games), the policy is learned from planning.
- i.e. The agent can “think ahead”, enumerating the return for every possible action (in the given state) for a given time horizon.
- The agent then executes the plan with the best return (=taking the immediate action associated with the plan) and discarding all future planned actions.

Interaction Loop of Agent

The agent is embedded in an environment.

The environment is characterized by a current state (or observation).

- The agent takes an action in the environment.
- Based on the action, the agent may or may not receive a reward.
- The agent's action has changed the environment.
- At the same time, the agent's action transitions the agent to a new state / observation in the environment.
- The agent then takes another action in the environment, based on the current state / observation … etc.

Learning

Learning can be achieved using a fixed model, or, no model at all.

- Model based: The agent has a model of the environment, and uses the model to learn the policy.
- Model free: The agent learns the policy by pure trial and error.
- The latter can be very “expensive”, since a large number of trials are usually needed => NOT sample efficient.
- Note: In practice, humans don’t need a large number of trials to learn.
- Hence, how do we “fast track” human based, model free learning to RL?

What is Prediction and What is Control?

- In the RL context, prediction is the evaluation of the policy = policy evaluation.

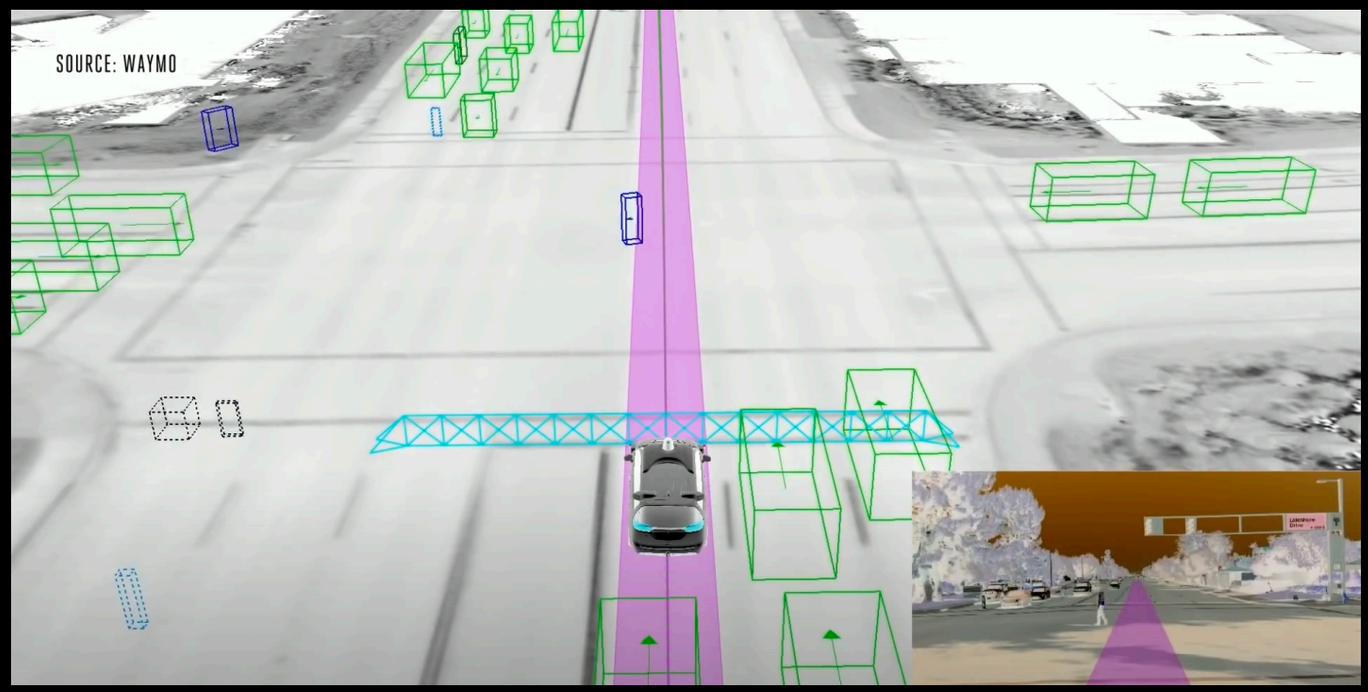
- i.e. How well will this policy perform?

- In the RL context, control is the optimization of the policy.

- In order to optimize a policy, you need to know how well the policy is currently performing.

- Therefore, prediction provides the necessary feedback to perform control.

What is the Role of Perception in Reinforcement Learning?



Not every RL problem involves / needs perception (pixel data).

Example: The cart-pole balancing problem only requires knowledge of the position, velocity, angle and angular velocity of the pole.

This knowledge can be acquired (directly) through sensors.

At the same time, these quantities can also be acquired using a sequence of video frames.

In general, many RL algorithms will benefit from perception (self driving cars, robot navigation, delivery drones, etc.) due to the rich information contained in an image / video frame(s).

What is the Role of Perception in Reinforcement Learning?

- Classical RL algorithms and their benchmarks, did not involve raw pixel data / images.

- Current state of the art RL algorithms such as Proximal Policy Optimization (PPO) and Soft Actor Critic (SAC) were developed using non visual state inputs (=without raw pixel data / images).

- Deep Q Network (DQN) was the first RL algorithm to successfully use pixel data / images, to train a policy.

- More recently, algorithms like DrQ and Dreamer have used raw pixel data / images in combination with an actor-critic algorithm, to achieve state of the art results.

How is Reinforcement Learning Different from Supervised or Unsupervised Learning?

- Reinforcement learning incorporates the concept of reward.
- Reward acts as a signal to encourage or discourage different decisions.
- Biologically, the human brain releases dopamine, in anticipation of receiving a reward.
- Hence, reward is an innate attribute that is hardwired into humans.

How is Reinforcement Learning Different from Supervised or Unsupervised Learning?

- In RL, an agent needs to interact with the environment.
- By interacting with the environment, the agent learns and dynamically changes / influences the environment.
- Unlike SL and UL, a temporal component is built into the RL problem, since sequential decision making is involved.
- Because of this, learning needs to be done differently.

How is Reinforcement Learning Different from Supervised or Unsupervised Learning?

- Loss functions are used in SL, UL and RL.
- In SL or UL, a decreasing loss function indicates that the model is improving.
- In RL, this is NOT true.
- In RL, an increasing (decreasing) expected return indicates that the policy is improving.
- In RL, the expected return needs to be computed for many, many seeds (say 10 seeds), in order to reach a definitive conclusion.

What's Next?

- The previous slides were meant to provide a simple introduction to reinforcement learning.

- In addition, explanations were provided to common "What?" and "How?" questions.

- Next, let's look at some fundamental building blocks.