Machine Learning Engineer Nanodegree

Capstone Proposal

Ann Lee
November 4, 2018

Domain Background

This proposal primarily applies reinforcement learning to the traveling salesman problem. Traveling salesman problem tries to find an optimal solution for the shortest path. Reinforcement learning enables an agent to learn in an interactive environment by trial and error using feedback from its own actions and experiences. There are some researches are mentioned to reinforcement learning can help traveling salesman problem. Below are some related academic researches

1. Reinforcement Learning: An Introduction 2nd Edition, Richard S. Sutton and Andrew G. Barto. Section 6.5, Q-learning: Off-policy TD Control.
2. Bello, Irwan, Pham, Hieu, Le, Quoc V, Norouzi, Mohammad, and Bengio, Samy. Neural combinatorial optimization with reinforcement learning. arXiv:1611.09940, 2016. Section 3, NEURAL NETWORK ARCHITECTURE FOR TSP.
3. Hanjun Dai, Elias B. Khalil, Yuyu Zhang, Bistra Dilkina, Le Song. “Learning Combinatorial Optimization Algorithms over Graphs”. arXiv:1704.01665, 2017. Section 6, Training: Q-learning. and so on.

Problem Statement

Based on "Santa's Stolen Sleigh" competition from the Kaggle which requirement is to optimize the routes Santa will take to, I tell another story. Santa is planning to deliver presents to children in Japan in Christmas night. He asks the shortest path. In addition, he wishes to train his reindeer friends to go by themselves then he can have tea time leisurely in the future.

Datasets and Inputs

The dataset is from "Santa's Stolen Sleigh" competition in the Kaggle which contains 100000 data and is a list of all gifts that need to be delivered. The

dataset includes 4 columns, gift id, latitude, longitude, and weight. In my project, I only grab the latitude and longitude columns. Using latitude and longitude to find the location of the destination.
Since my project only focuses on a small map, I will filter out the destination which only located in Japan as my list to deliver gifts. Each record means a destination of a gift to deliver and there is a path between any two of destinations. The purpose is trying to find out a close optimal shortest path.

Solution Statement

At first, the dataset needs to be processed. Google maps API should help. According to the latitude and longitude to recognize the location of the gift list. After the dataset is ready, the problem can be treated as the traveling salesman problem (TSP). Reinforcement learning helps the shortest path with the greedy policy. Furthermore, using negative tour length as the reward.

Benchmark Model

In this project, I choose the greedy algorithm as the benchmark model. A greedy algorithm lets the salesman choose the nearest unvisited city as his next move. This algorithm quickly yields an effectively short route. After obtaining an optimal shortest path can calculate the total distance of tours.

Evaluation Metrics

The benchmark model, the greedy algorithm for traveling salesman problem, after finding the shortest path and count the total distance of tours will be based to compare to the result from reinforcement learning Q-learning. Using the reinforcement learning, every episode will get a reward which is equal to the minus of the total distance of tours. Calculating the total distance for traveling to evaluate how well reinforcement learning does.

Project Design

At first, reduce the scope of the map. Then implements the traveling salesman problem using Q-learning.

- States: a state S is a sequence of actions on a graph G.
- Actions: an action A is a node of G that is not part of the current state S.
- Rewards: minus of distance is the reward.
- Pseudocode:

Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(\mathit{terminal}, \cdot) = 0$

Loop for each episode:
  Initialize S
  Loop for each step of episode:
    Choose A from S using policy derived from Q (e.g., ε-greedy)
    Take action A, observe R, S'
    $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma \max_a Q(S', a) - Q(S, A)\big]$
    $S \leftarrow S'$
  until S is terminal