Methods for Autonomous Path Planning

Abhi Kulgod, Anthony Degleris, Isaac Scheinfeld

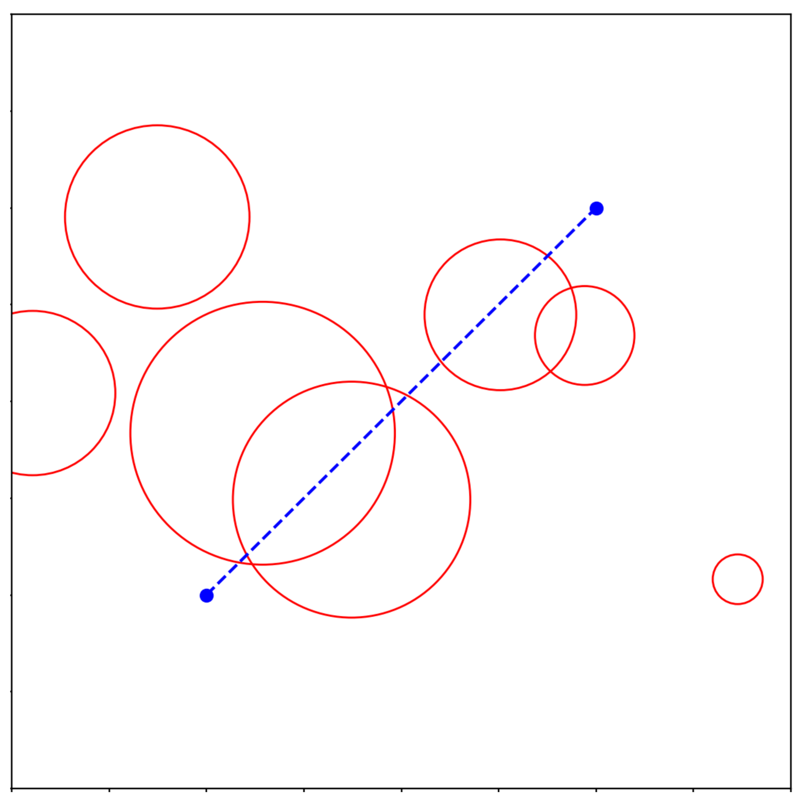
Motivation

- Path planning is the task of finding the optimal route between two points, avoiding any obstacles along the way
- This task arises naturally in many vehicle control tasks. Our approach is modeled after drone navigation, where the goal is to find a shortest path.
- Classical techniques often rely on simplifying assumptions and require a new solution each time the environment changes
- A solution that makes fewer assumptions and generalizes well over different environments is preferable
- Reinforcement learning techniques can implicitly learn a general solution, and have more potential for extension to dynamic environments

Problem Setup

- Randomly generated with 1-8 circular obstacles
- Rejected obstacles covering endpoints.
- **Goal:** Plan a path within the boundary and outside of the obstacles (path is a sequence of unit spaced points)

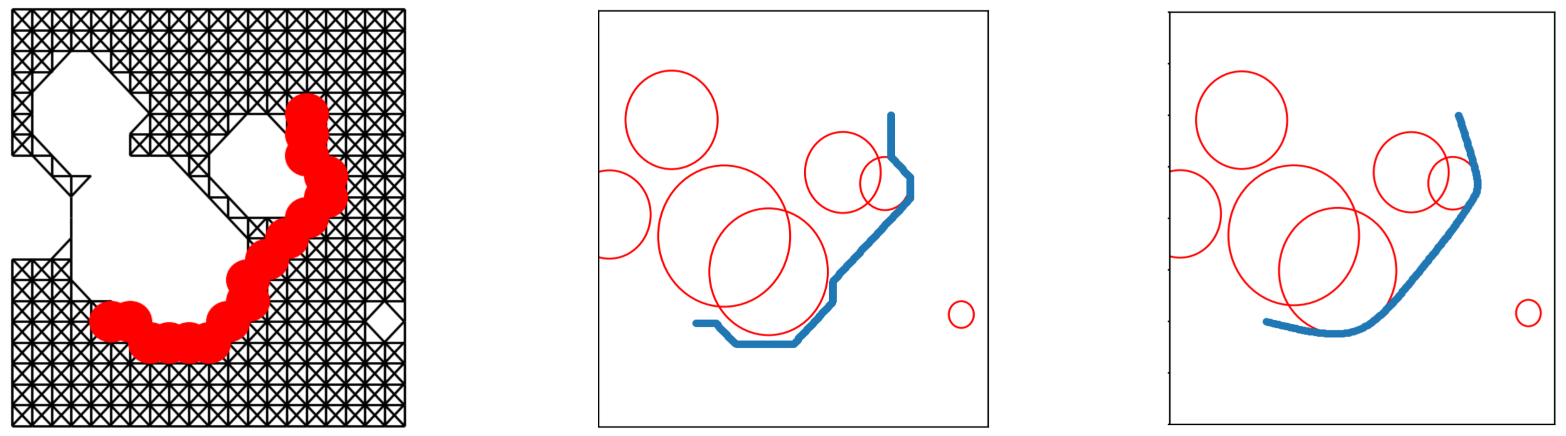
$$\begin{aligned} \text{Boundary} &= [-50, 150]^2 \\ \text{Start} &= (0, 0) \\ \text{End} &= (100, 100) \\ \text{Radius} &\sim \mathcal{N}(25, 10) \end{aligned}$$



Methods

Graph Approximation and Optimization[1]

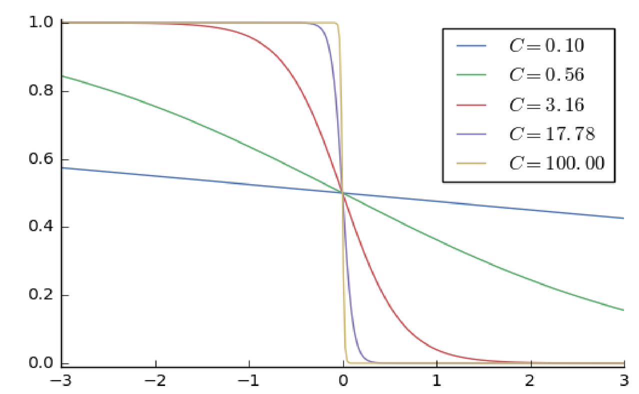
- Approximates the best path by finding the shortest path in a space-filling graph



- Uses optimization (gradient descent) to stretch and smooth, minimizing

$$\mathcal{L}(x; c, R, C) = \sum_i \left[\sum_j \text{sigmoid}\left(-C\left(\frac{\|x_i - c_j\|_2^2}{R_j^2} - 1\right)\right) + \eta\|x_i - x_{i+1}\|_2^2\right]$$

- Sigmoid represents obstacle penalty. As gradient descent converges, we send $C \to \infty$ hardening the boundary.



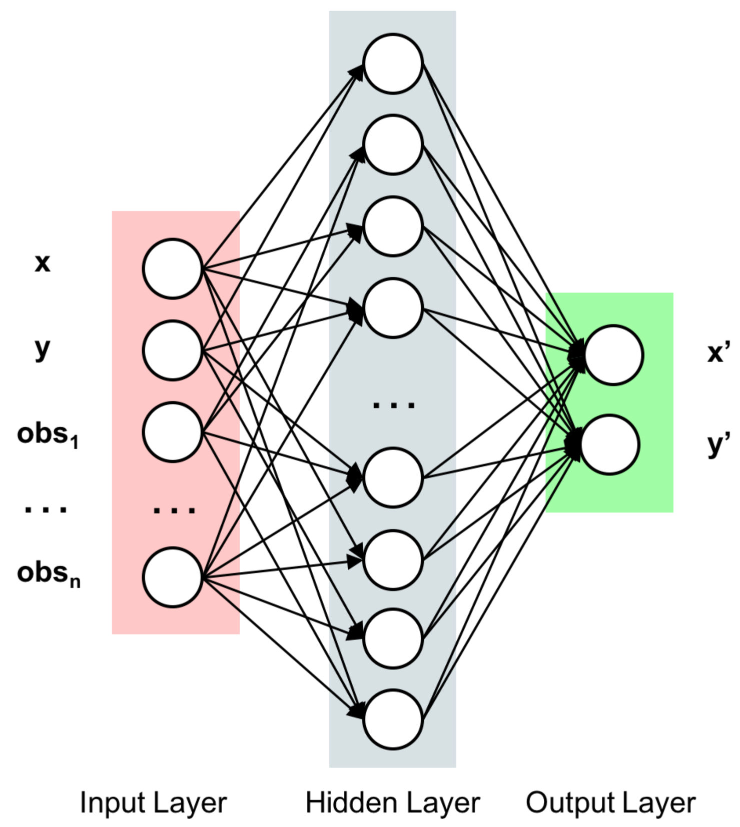
Fitted Policy Iteration (FPI)

- Fit a model that predicts the path in general (not for a specific map)
- State space is continuous and high-dimensional
- Value function would be extremely complicated
- Instead, estimate the policy directly

$$\pi(s) = \theta^T \phi(s)$$

- Update rule

$$\pi^{t+1}(s) \approx \arg\max_a \sum_{s' \in S} P_{sa}(s') V^{\pi^t}(s')$$

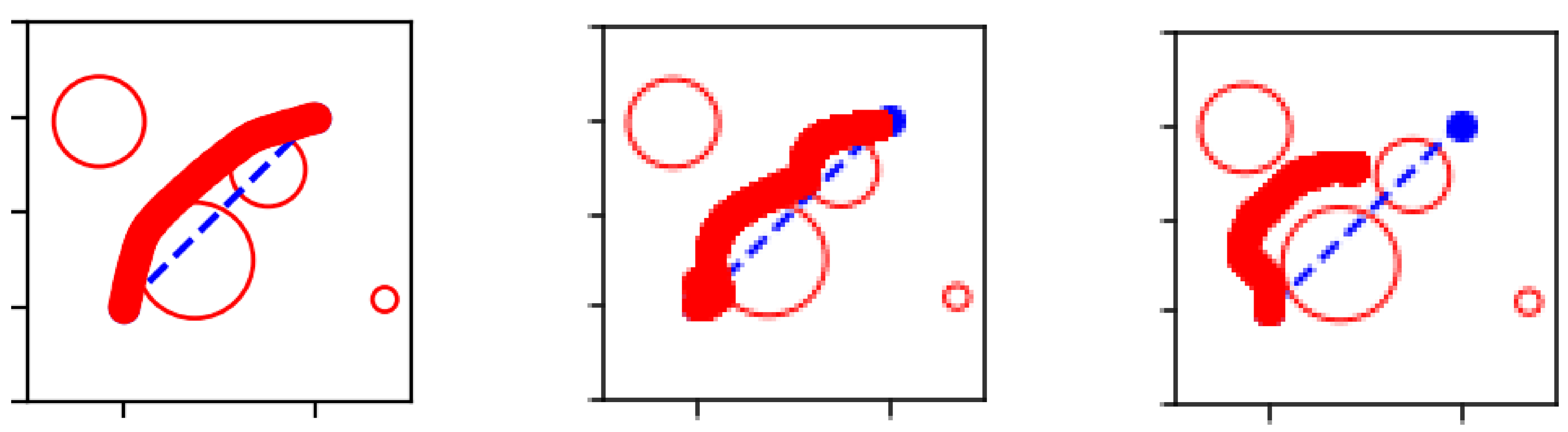


Neural Net Architecture

Deep Fitted Policy Iteration (DFPI)

- Use a single layer feed forward neural network to improve policy approximation

Paths generated on one test map (DFPI fails)

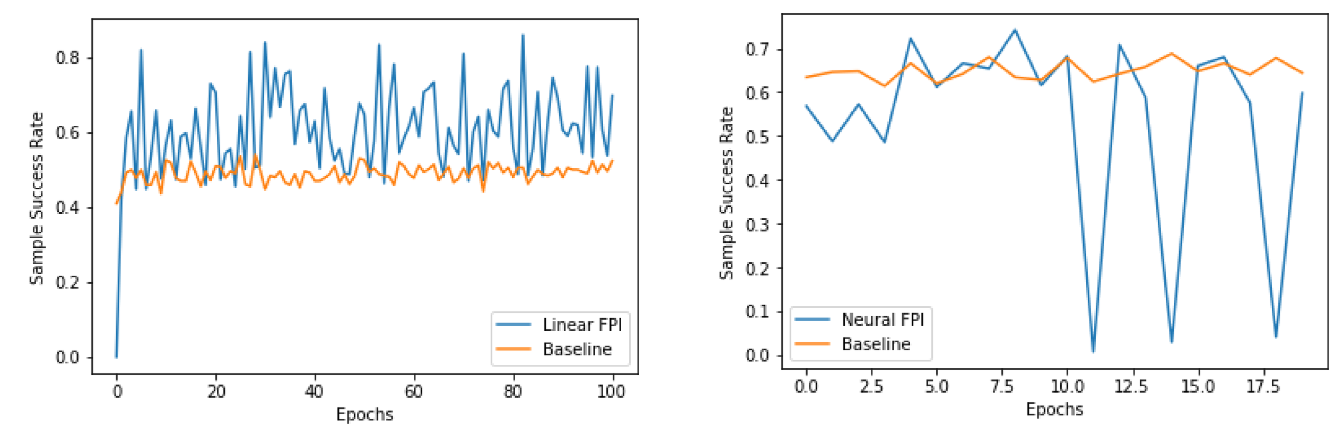


Optimization FPI Deep FPI

Results

Algorithm	Success Rate	Average Path Length
Optimization	1.0	156.4
FPI	0.88	221.4
DFPI	0.31	861.7

Learning Curves



Discussion

.

- We could not train any fitted value iteration model to generate a meaningful model
  - WHY
- The FPI model achieved surprising success
  - General models for path planning are possible
  - Direct policy approximation can replace value approximation in some problems
- The DFPI model underperformed significantly
  - Our neural network architecture failed to capture the feature complexity

Future Work

- Improve NN architecture to capture relevant features
- Address more complex problems (moving obstacles, path smoothness constraints, etc.)
- Explore fitted policy iteration further, comparing it to fitted value iteration on known problems

References

1. [1] G. Angeris, "Some thoughts on global path optimization", 2017. [Online]. Available: https://guille.site/path-optimization-thoughts.html. [Accessed: 11- Dec- 2017].
2. [2] K. Gregory, V. A, V. Pong and P. Abbeel, "Uncertainty-Aware Reinforcement Learning for Collision Avoidance", 2017.