Learning to Search for Targets

with Deep Reinforcement Learning

Oskar Lundin

Linköping University

April 19, 2022

Outline

Problem Description

Autonomous search for a set of targets in an environment with a fixed camera.

- Agent observes a limited region of the environment.
- Can direct its gaze and indicate when a target is in view through actions.
- Should locate all targets while minimizing the number of actions.
- Applications in search and rescue, fire detection, surveillance, etc.

- In a random environment with uniformly distributed targets, random search is sufficient.
- Most real-world search tasks are not random, but exhibit structure.
- Cues in the searched scene can be used to find targets quicker.
  - Books are in bookshelves.
  - Cars can be found on roads.
  - Some targets spread out
  - Some are close together.
- Patterns and cues may be subtle and difficult to pick up.
- Manually engineering a searching system with rules can be difficult and costly.
- Can a system learn to search intelligently from a set of samples and generalize to similar search tasks?

Challenges

- Prioritize regions with high probability of targets based on previous experience.
- Find multiple targets while minimizing path length.
- Search exhaustively while avoiding searching the same region twice.

Research Questions

1. How can an agent that learns to intelligently search for targets be implemented with reinforcement learning?
2. What is a suitable memory architecture for a visual search agent?
3. How does the learning agent compare to random walk, exhaustive search and a human searcher with prior knowledge of the searched scene?
4. How does the agent's ability to generalize to unseen in-distribution environments depend on the number of training samples?

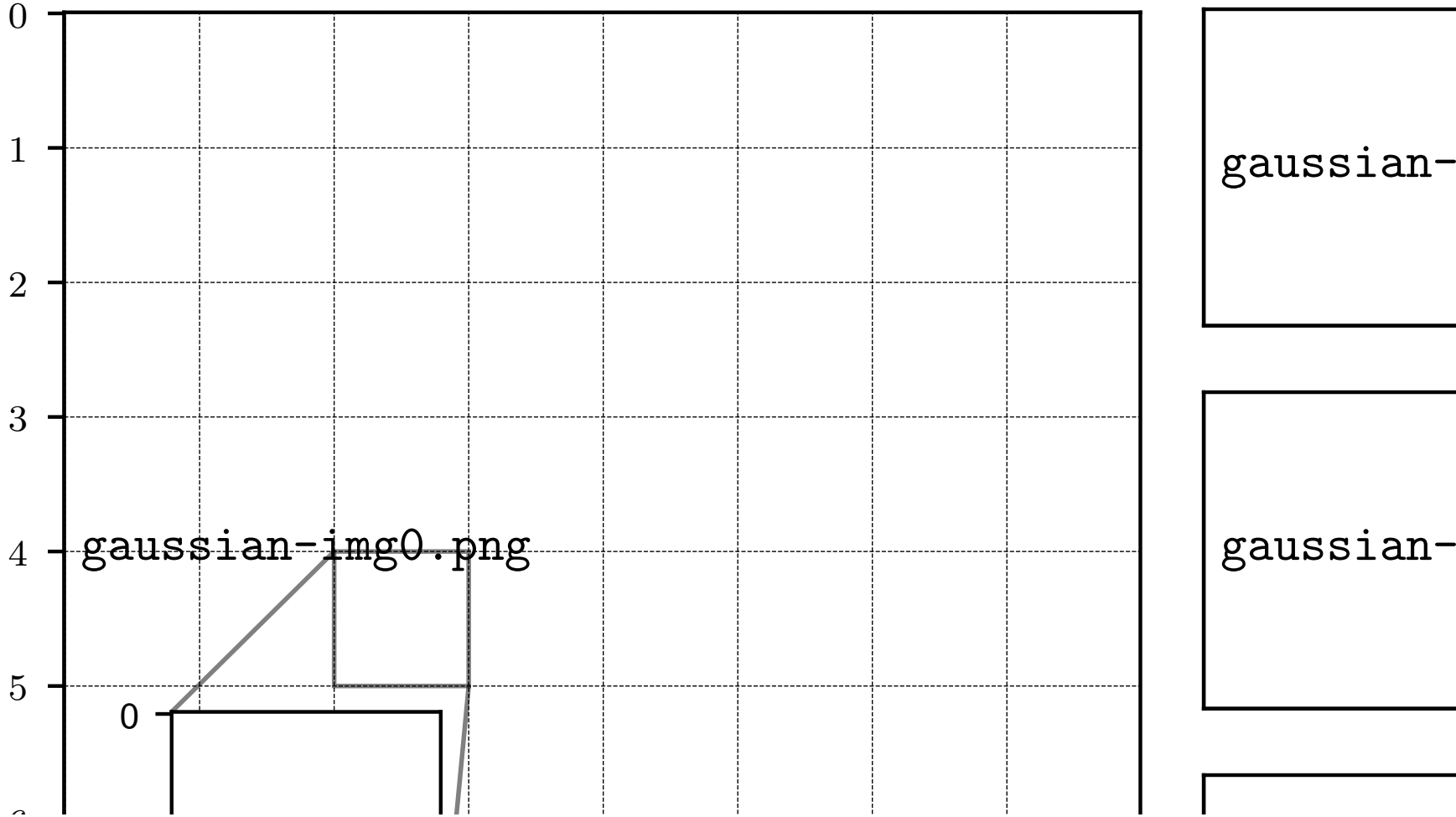
Reinforcement Learning

Reinforcement learning (RL) is a paradigm for learning mappings from observations to actions.
A common

Method

Environments

Gaussian Environment



Observation, Actions and Reward

Agent

Implementation

- Environments implemented in OpenAI gym.
- Models implemented and trained with PyTorch.

Preliminiary Results

Problems:

- Long training times.
- Hyperparameter tuning expensive.
- Difficult to predict how a change impacts results.
- Design of reward signal.
- Collecting results over many seeds will take time...
- More or less expected results for small search spaces (8×8 positions).
- Convergence too slow with baselines for large search spaces (16×16 positions).
- Spatial memory approach can handle it.

Future Steps

1. Fix hyperparameters and other design decisions.
2. Collect complete results across multiple seeds.
3. Compare to baselines and non-learning agents.
4. Evaluate generalization.
5. Discussion and conclusion.
6. Presentation preparation.