Learning Machines Demo 3

Catching the prey

Team α - Mick IJzer, Leon Berberich, Justus Huebotter & Nikolay Polyanov

Agenda

- Problem definition
- Methodology
- Experimental setup
- Results
- Conclusions

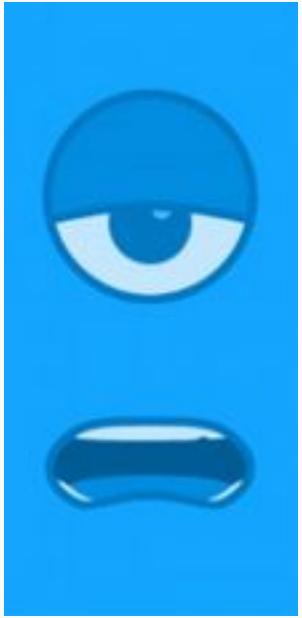
Task & Problem Definition

- Task:
    - **Hunting down the prey** (hunter) and **avoiding the hunter** (prey) with the Robobo
    - Robobo should be able to **detect its target** of the desired color in the arena and change it's path in order to **catch/avoid** it
- Problems:
    - **Color sensitive object recognition** using **camera** of the mobile phone
    - (Population) **learning from experience**
        - **Obstacle avoidance**
        - **Catch** the prey as fast as possible (hunter)
        - **Survive** as long as possible (prey)

Problems Faced

There were several challenges faced while working on the task:

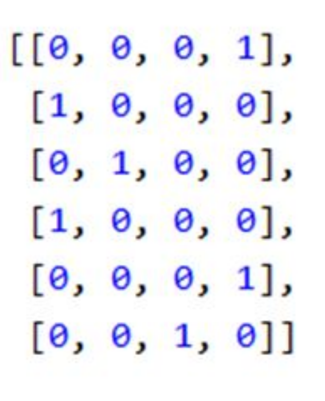
- Running the same script for two robots at the same time
- Difference in cameras between the mobile phones used
- Restart of experiments limited time
  - Took gatherer (last week) as hunter
  - Only evolved prey!



Methods Overview

A combination of an Evolutionary Algorithm and Policy Iteration:

- Learning algorithm:
  - Evolutionary Strategy ($\mu + \lambda$)
  - Variation (crossover & mutation)
  - Selection (probabilistic parent & survival selection)
- Controllers:
  - Deterministic policies π
  - Each row represents a state
  - Each column represents an action
  - Deterministic → Action *a* taken in some state *s* will always be the same for an individual

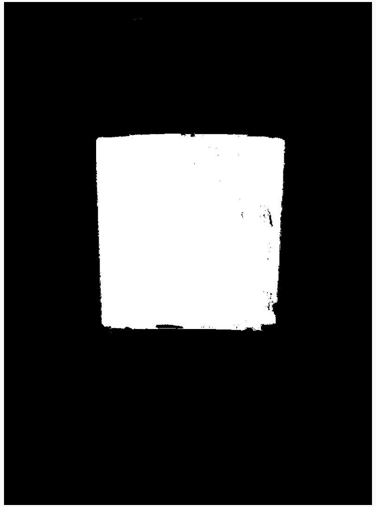


```
[[0, 0, 0, 1],
 [1, 0, 0, 0],
 [0, 1, 0, 0],
 [1, 0, 0, 0],
 [0, 0, 0, 1],
 [0, 0, 1, 0]]
```

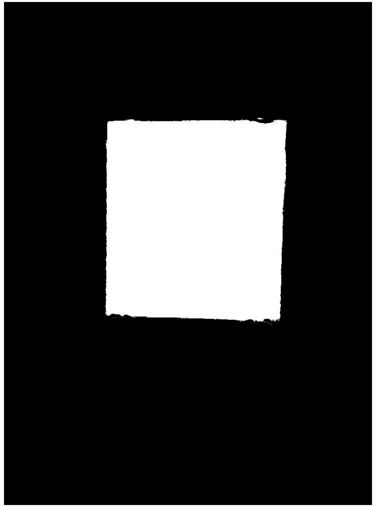
Image Processing

- Cropped image
- OpenCV 2 - findContours
- Red in range (BGR)
  - [0, 0, 180] - [150, 150, 255] for hunter
  - [0, 0, 180] - [80, 80, 255] for prey

Hunter view



Prey view



Controller Representation

Actions:

- Forward
- Left turn
- Right turn

Bits included in state:

- IR sensor: object detected right or front
- IR sensor: object detected left
- Camera: target detected
- Memory: target seen in recent iterations
- Memory: target last seen on left or right

Deterministic policy π with 3 columns and 2^5 = 32 rows.

There are 3^32 = 1.85 × 10^{15} possible combinations

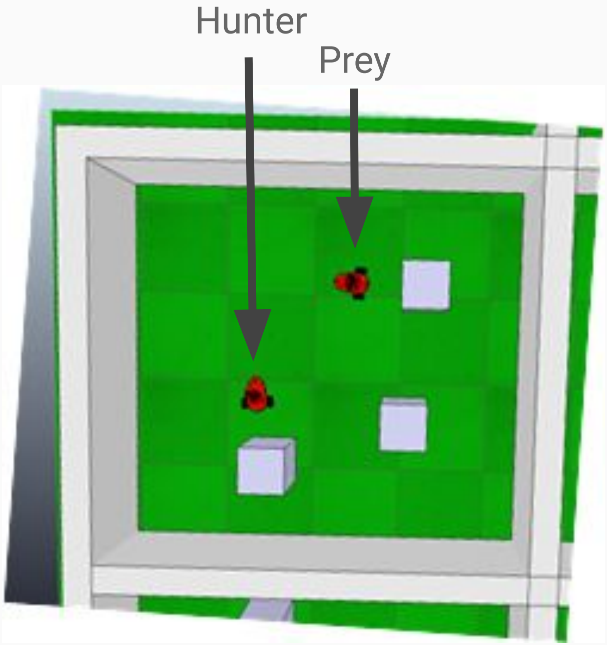
Evolution Parameters & Metrics

Phenotype	Behavior of the robot when in a state
Genotype	Deterministic policy (where every row is a state with 0 or 1 for each action)
Mapping	Readings & policy table are translated into robot movement by a python function
Fitness	+1 per survived timestep, -50 getting caught, -50 on wall hit (100 steps in total → max fitness = 100)
Crossover	Uniform (Every row is a copy from either P1 or P2)
Mutation	Shuffle a row with probability mu = 0.1
General Settings	Parent selection: prob. rank (2 parents) Survivor selection: prob. rank (10 individuals) Initialization: random Termination: 11 generations

Experimental Setup

In simulation obstacles were added, size and floor color are also different:

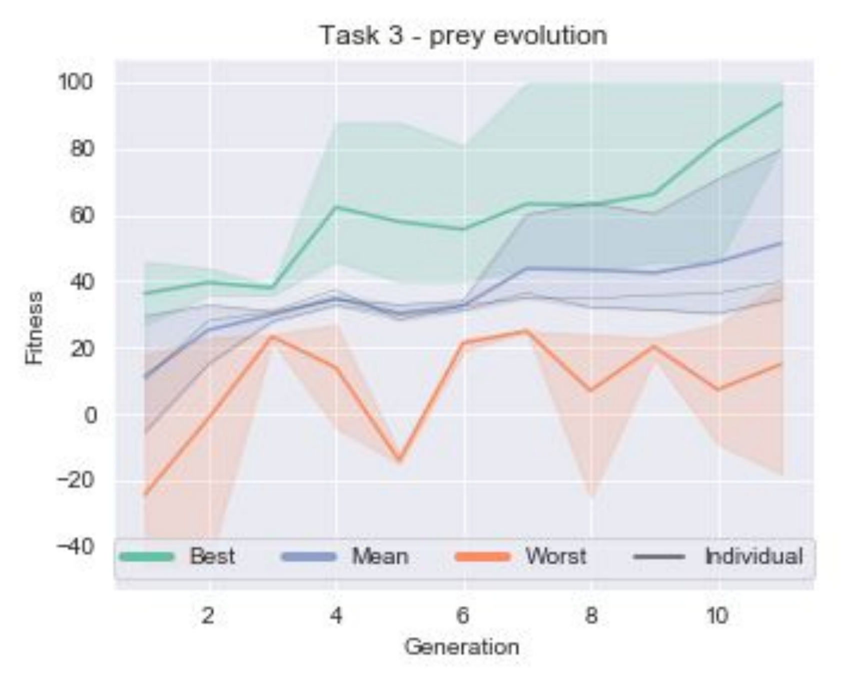
- Hunter and prey are both red
- Position of the robots and size of the arena (2m x 2m) remain constant.
- Position → faster learning of obstacle avoidance



Results - Population

- 11 generations
- 10 individuals per generation
- 3 reruns
- Multiple best individuals

⅔ reruns performed really bad



Reflection Task & Problems

- Task:
    - **Hunting down the prey** (hunter) ✓
    - **Avoiding the hunter** (prey) ✓
    - Robobo should be able to **detect its target** of the desired color change it's path in order to **catch/avoid** it ✓
- Problems:
    - **Color sensitive object recognition** using **camera** of the mobile phone ✓
    - (Population) **learning from experience** ✓
        - **Obstacle avoidance**
        - Catch the prey as fast as possible (hunter) ✓
        - Survive as long as possible (prey)

Conclusions

- Evolutionary Approach gives a promising learning curve for the population
- Prey behavior seems to be more complex than predator behavior
- Best individuals exhibit desired behavior
- What we would have liked to try with more time:
    - Co-evolution of hunter and prey populations
    - Increased state space (i.e. back sensors)
    - Multiple scenes (i.e. stochastic)
    - “League” setup

Thank you for your kind attention!

Questions?
Demonstration!