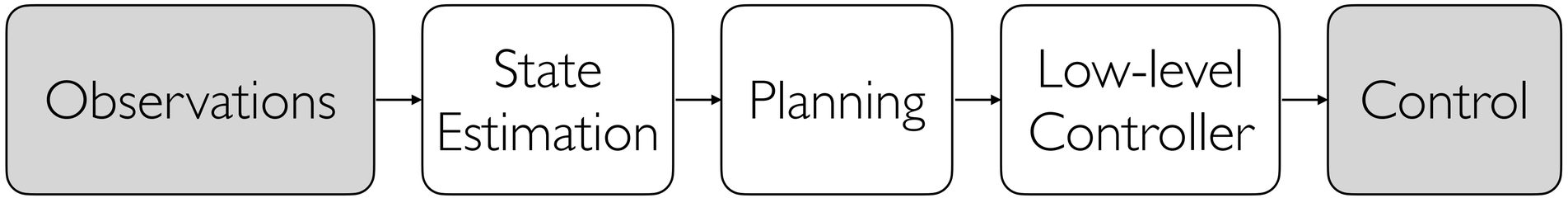
Markov Decision Processes and Deep RL

Typical Robotics Pipeline



Do we need to have this pipeline, or could I just directly map pixels to actions?

- Error propagation from one module to the next
- Difficulty in defining state (e.g. chopping onions)
- (sometimes) Easier to map to control than to do explicit state estimation

Policies may be simpler

Journal of Experimental Psychology:
Human Perception and Performance
1996, Vol. 22, No. 3, 531–543

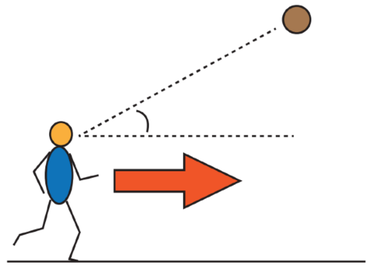
0096-1523/96/$3.00

Do Fielders Know Where to Go to Catch the Ball or Only How to Get There?

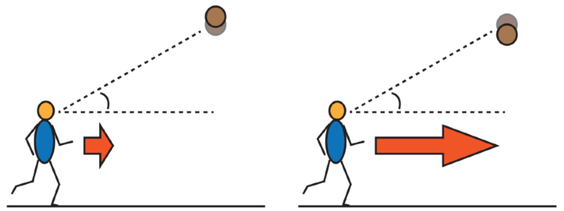
Peter McLeod
Oxford University

Zoltan Dienes
Sussex University

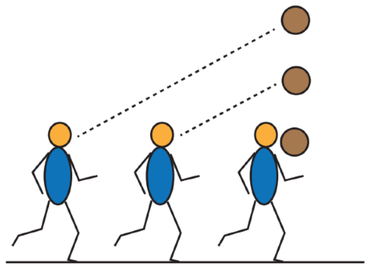
Skilled fielders were filmed as they ran backward or forward to catch balls projected toward them from a bowling machine 45 m away. They ran at a speed that kept the acceleration of the tangent of the angle of elevation of gaze to the ball at 0. This algorithm does not tell fielders where or when the ball will land, but it ensures that they run through the place where the ball drops to catch height at the precise moment that the ball arrives there. The algorithm leads to interception of the ball irrespective of the effect of wind resistance on the trajectory of the ball.



Modulate running speed to maintain angle between ball and ground.



If ball rises in the field of view, slow down.
If ball drops, speed up.
Thus, position of the ball in the field of view is maintained.



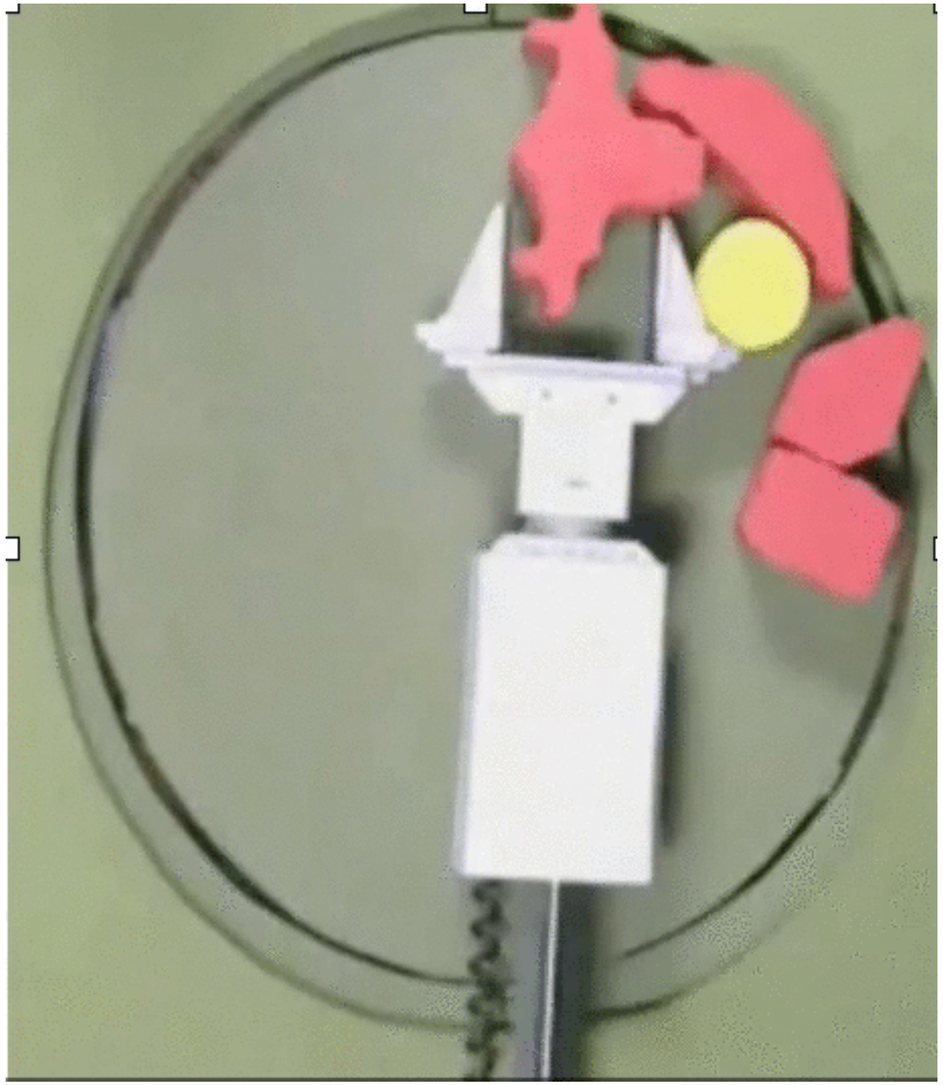
Using this heuristic, human catcher arrives at landing point exactly when the ball lands.

Do fielders know where to go to catch the ball or only how to get there? P. McLeod, Z. Dienes

Image Credit: Sergey Levine

Robot Learning vs Visual Learning

- Supervision
  - More than one answer
  - Delayed
- Non-stationarity
- Exploration vs exploitation
- …



Agent Environment Interface

Image Source: https://www.freecodecamp.org/news/a-brief-introduction-to-reinforcement-learning-7799af5840db/

Markov Decision Process

o_t

Step Back

a_t

o_{t+1}

...

Transition Function

How you move,
how the tiger moves

Reward Function

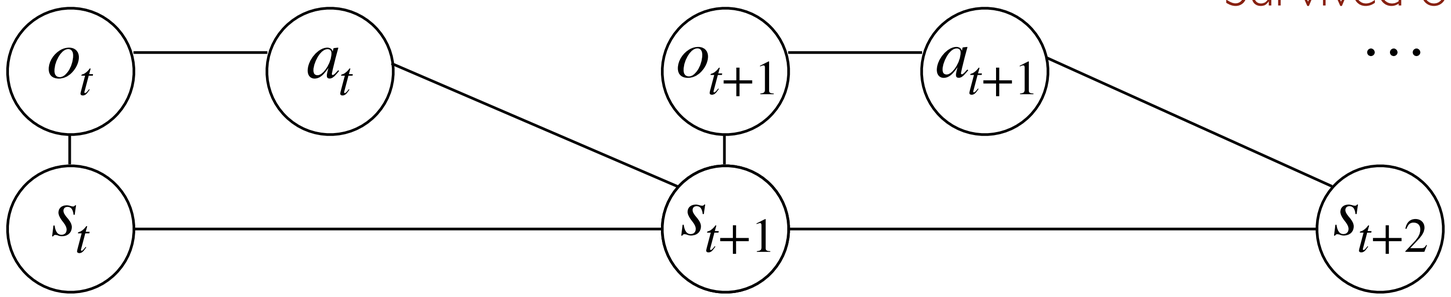
Survived or not

...

3D Relative Pose

s_t

s_{t+1}



One step dynamics $p(s_{t+1}, r_{t+1} \mid s_t, a_t)$

Transition Function $p(s_{t+1} \mid s_t, a_t)$ $\quad p(s_{t+2} \mid s_{t+1}, a_{t+1})$

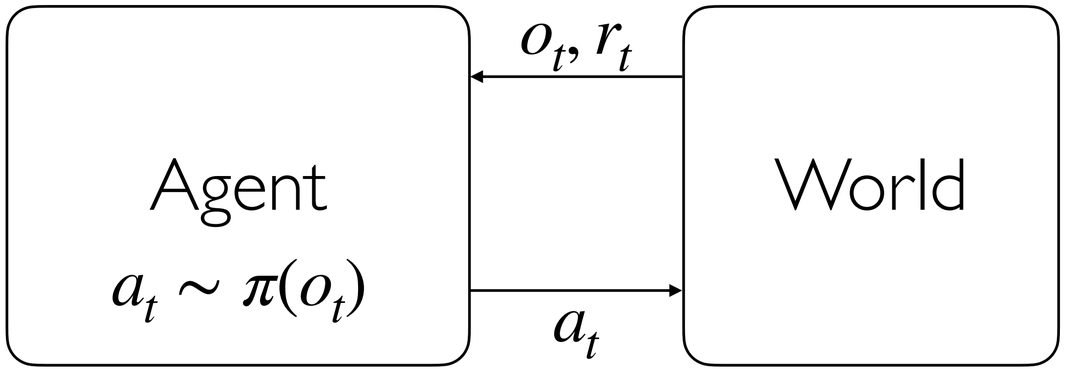
Reward Function $r_{t+1} = R(s_{t+1}, s_t, a_t)$ $\quad r_{t+2} = R(s_{t+2}, s_{t+1}, a_{t+1})$

Goal $\text{argmax}_{a_0,\ldots,a_T} \sum_t \gamma^t r_t$

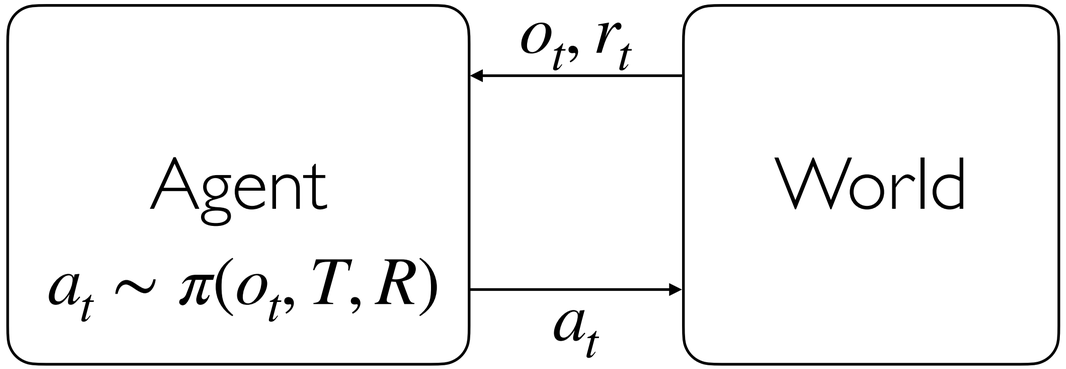
Solving MDPs

Policy: $a_t \sim \pi(o_t)$

Most General Case



More Specific Case



Fully Observed System $o_t = s_t$

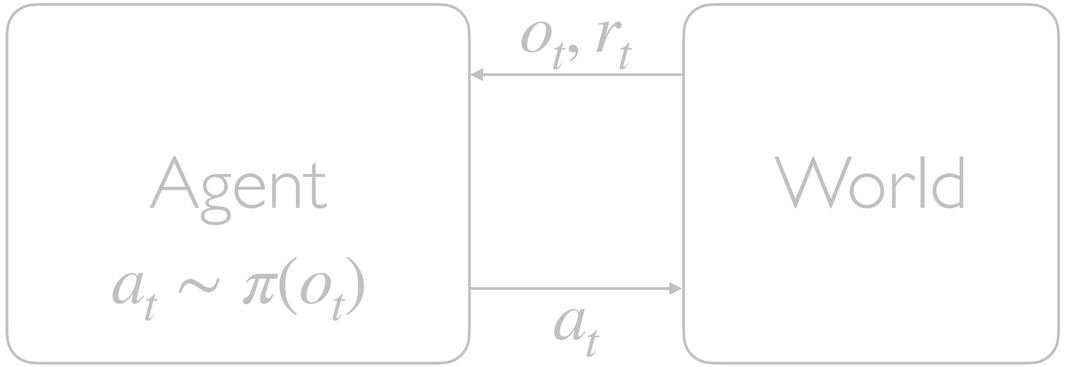
Known Transition Function $s_{t+1} \sim T(s_t, a_t)$

Known Reward Function $R(s_{t+1}, s_t, a_t)$

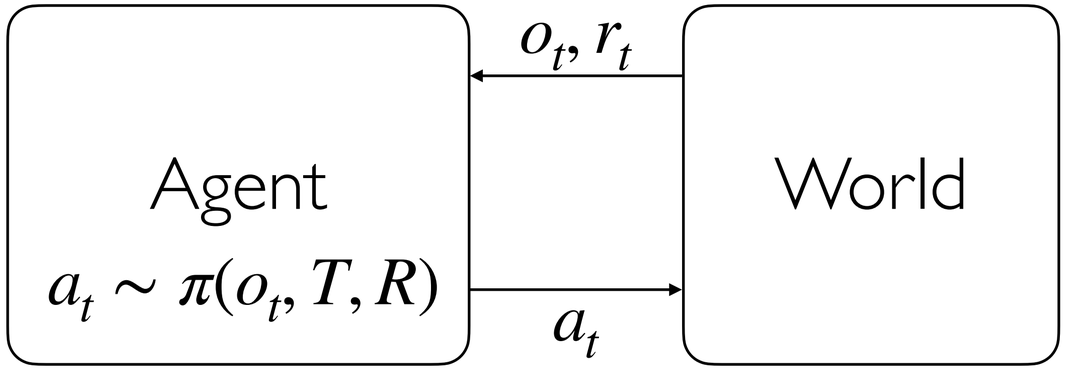
Solving MDPs

Policy: $a_t \sim \pi(o_t)$

Most General Case



More Specific Case



Fully Observed System $o_t = s_t$

Known Transition Function $s_{t+1} \sim T(s_t, a_t)$

Known Reward Function $R(s_{t+1}, s_t, a_t)$

Topics we will cover

- Basics
  - Definitions
  - Bellman Equations
- Solving known MDPs
  - Policy Evaluation, Policy Improvement, Policy Iteration, Value Iteration
- Solving unknown MDPs
  - Model-free policy evaluation
  - Model-free control
    - Q-learning
    - Policy Gradient