Reinforcement Learning for Autonomous Driving - Week 4

Brandon Dominique, Hoang Huynh, Eric Av, John Nguyen

- Meta-learning: John Nguyen
- Gazebo: Eric Av
- Reinforcement Learning: Hoang Huynh
- Meta-Cognitive Radio: Brandon Dominique

Meta-learning: Learning to Learn

- Introduction
- How/why it is useful
- Basic Example

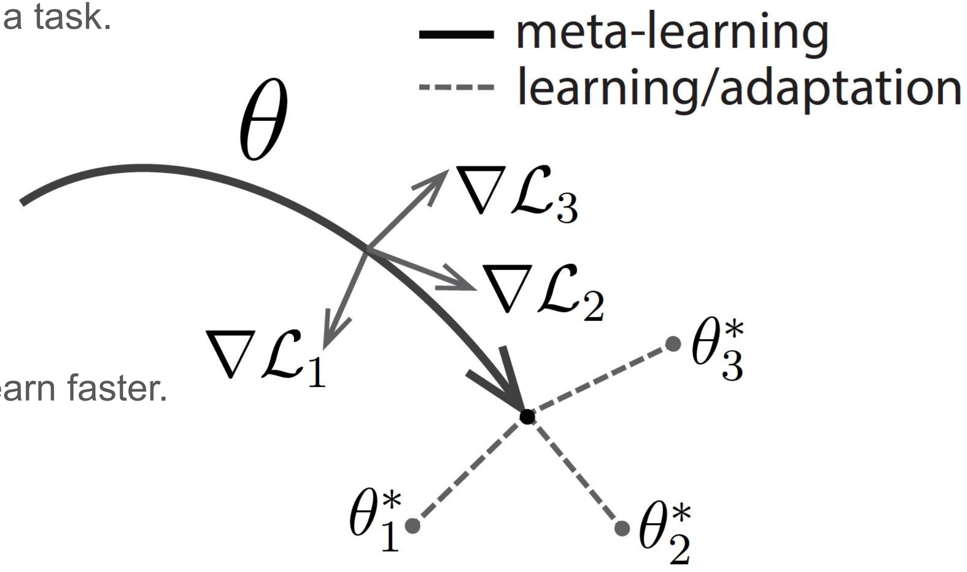
Algorithm 3 MAML for Reinforcement Learning

Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters

randomly initialize θ
while not done **do**
    Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
    for all $\mathcal{T}_i$ **do**
        Sample K trajectories $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{a}_1, ...\mathbf{x}_H)\}$ using f_θ in $\mathcal{T}_i$
        Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ using $\mathcal{D}$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation 4
        Compute adapted parameters with gradient descent: $\theta'_i = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
        Sample trajectories $\mathcal{D}'_i = \{(\mathbf{x}_1, \mathbf{a}_1, ...\mathbf{x}_H)\}$ using $f_{\theta'_i}$ in $\mathcal{T}_i$
    end for
    Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta'_i})$ using each $\mathcal{D}'_i$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation 4
end while

Meta-learning Framework: 2 AI's, 1 System

- Type of machine learning
  - Goal is to make a "model" very good at a task.
  - Ex. Fake vs. Real Image Recognition



- 2 Intelligences:
  - A "model" that learns the task.
  - An "agent" that changes the model to learn faster.
    - Changes parameters of model.

Learning a Class of Tasks

- Meta-learning methods teach a class of tasks to a model, instead of a single task.
    - Meta-learning lends itself well to few-shot learning: learning with a very small sample size.
- The goal of meta-learning is to teach general intelligence.
    - We want the model to be able to perform similar tasks well without training on each task individually.
        - Ex. recognizing fake dog pictures vs recognizing fake human pictures.

Toy Example: MAML + 2 Armed Bandit

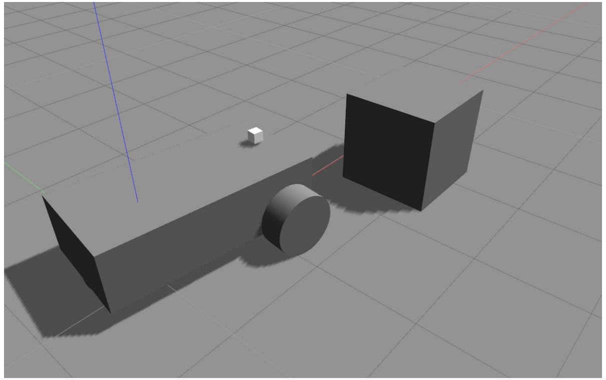
- 2 armed bandit problem: 2 slot machines with unknown probabilities to win.
  - Maximize Payout, Minimize Payout
- MAML is the gold standard meta-learning algorithm.
- Do not assume a slot machine is better than the other with no data.
  - I introduce a bias, and see if the agent can auto correct the bias.
- Framework:
  - The model attempts to figure out which of the 2 slot machines is better.
    - Problem: The model has a bias as to which slot machine is best without any data.
  - The agent attempts to correct this bias.

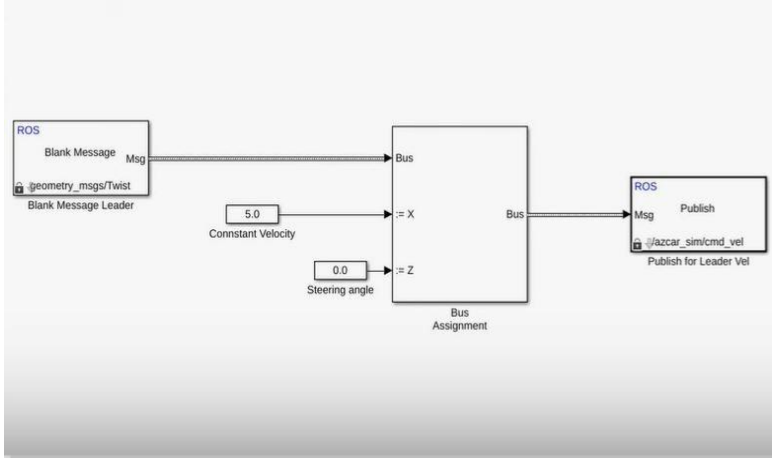
```
initial bias: [0.389 0.611]

final bias: [-0.001 -0.001]
```

ROS and Gazebo Successes

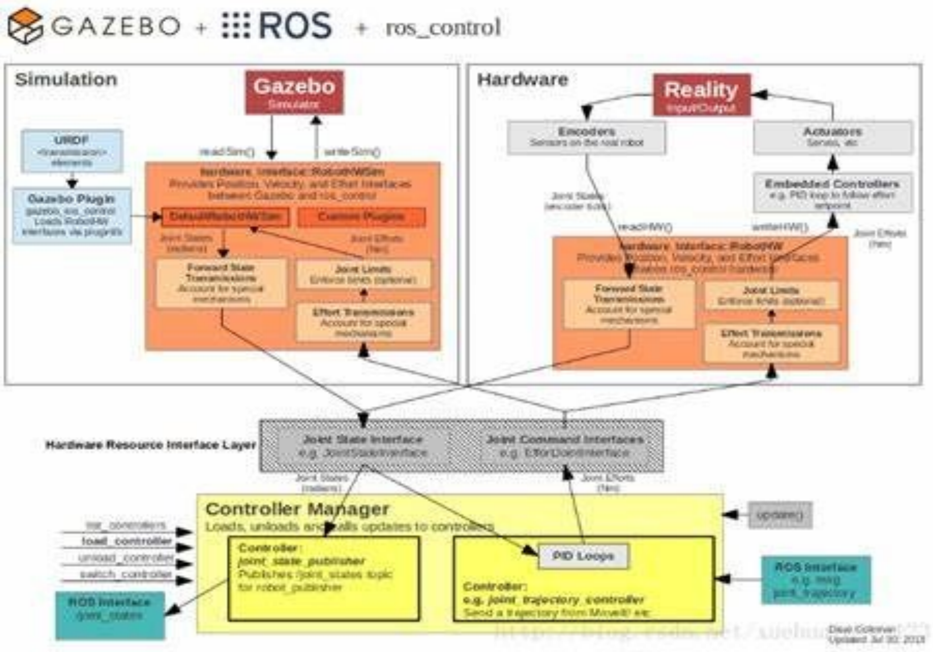
- Studied ROS and Gazebo relationship and importance
- Followed Car movement and forward sensor tutorial on Gazebosim
- Created first workspace with ROS and Gazebo to create a basic vehicle
- Used Simulink to command movement of car, essential for further implementation ideas





ROS and Gazebo Struggles

- Tutorials help but outdated tutorials are often misleading
- Troubles creating first workspace with ROS and Gazebo (again with the folders and different extensions ie: .launch, .world., .urdf or .sdf)
- Miscellaneous struggles concerning Ubuntu and console command learning curve



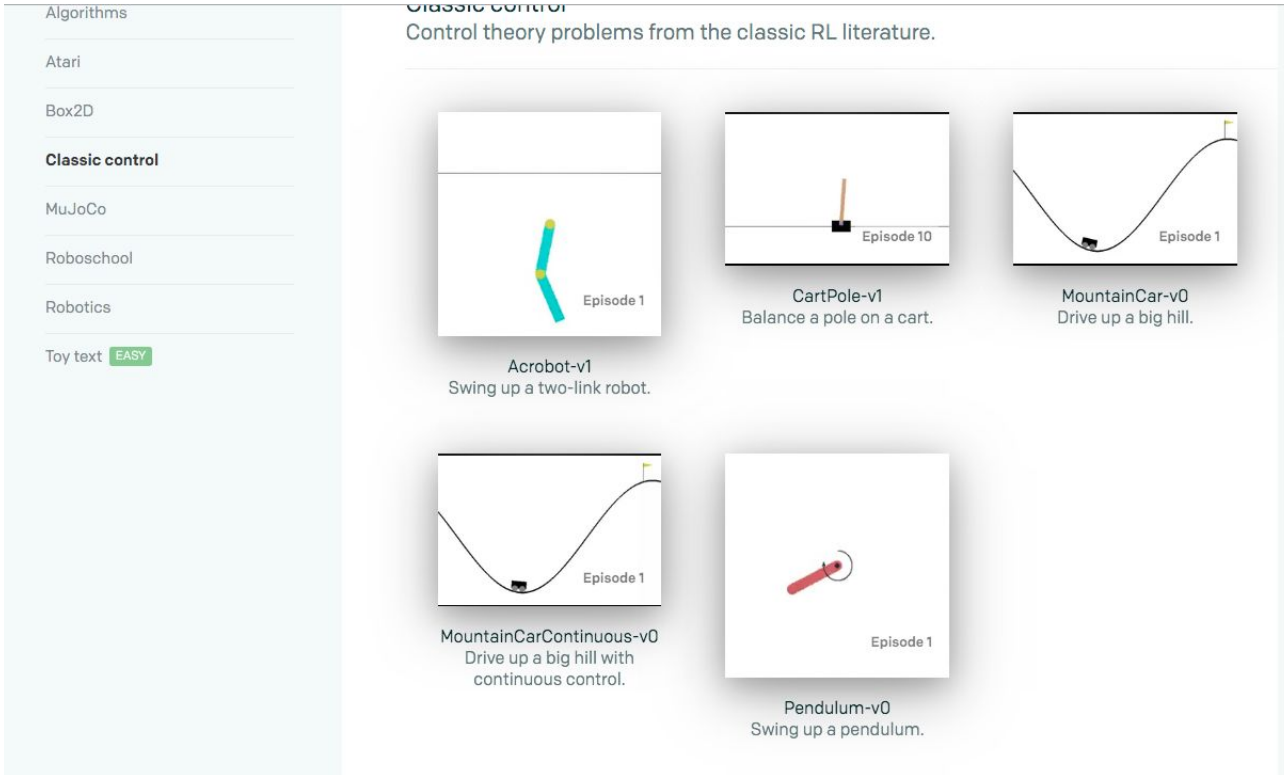
Apply Reinforcement Learning

- A non-profit research company
- Aims to promote and develop friendly AI in such a way as to benefit humanity

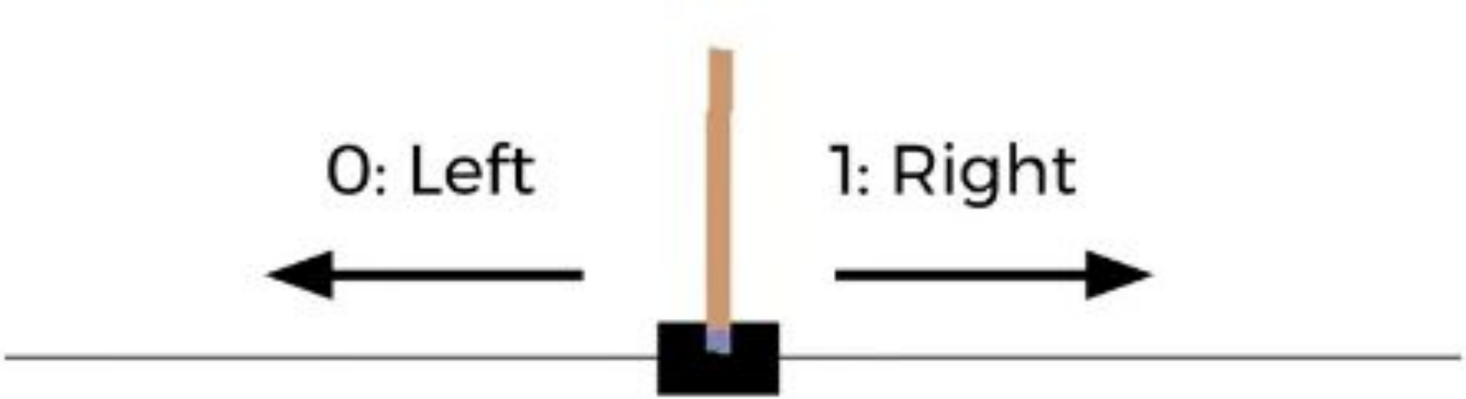
Apply Reinforcement Learning

Open AI gym

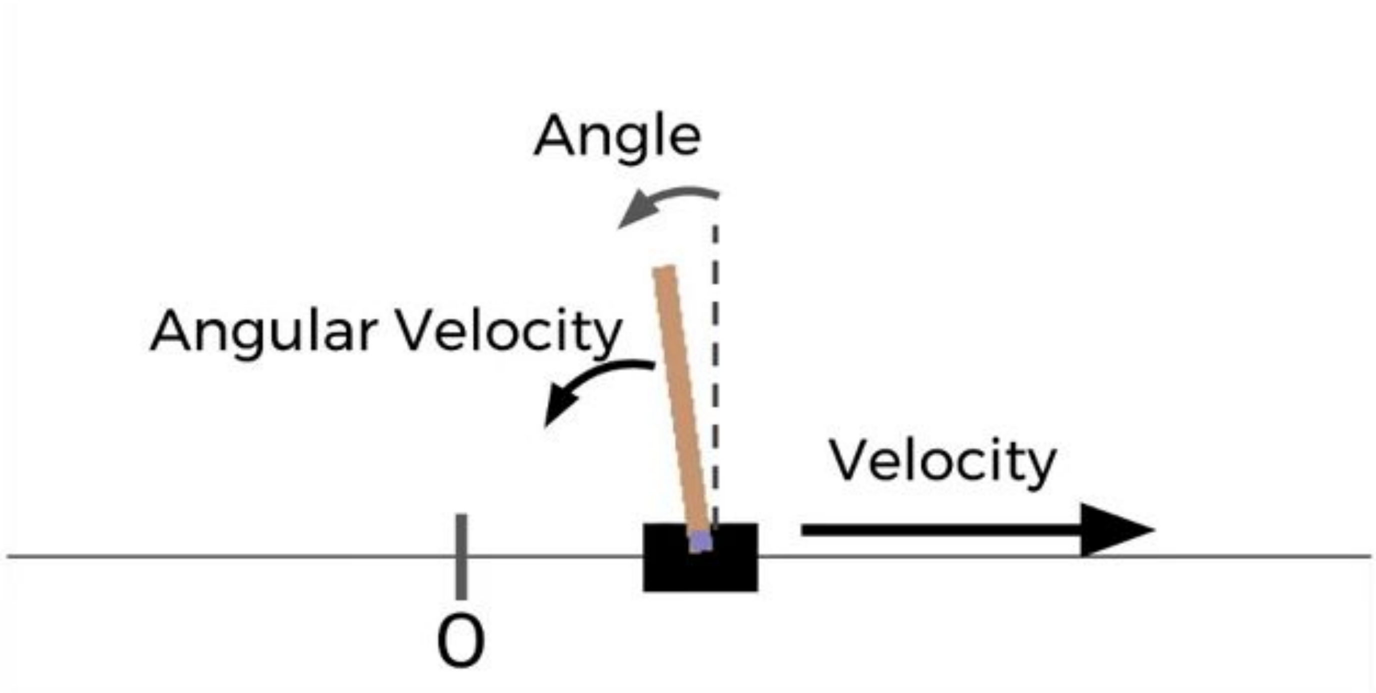
- a library that aids to develop and comparing reinforcement learning algorithms.



Cart Pole Game



Cart Pole Game

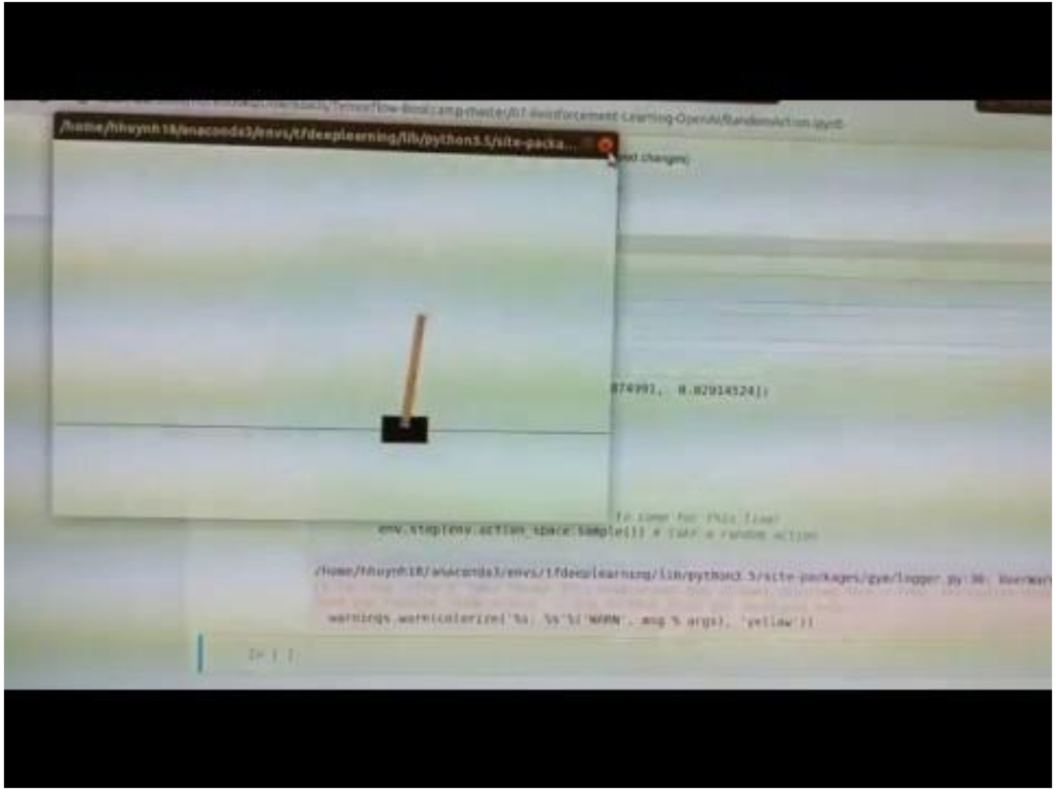


Policy Gradient Theory

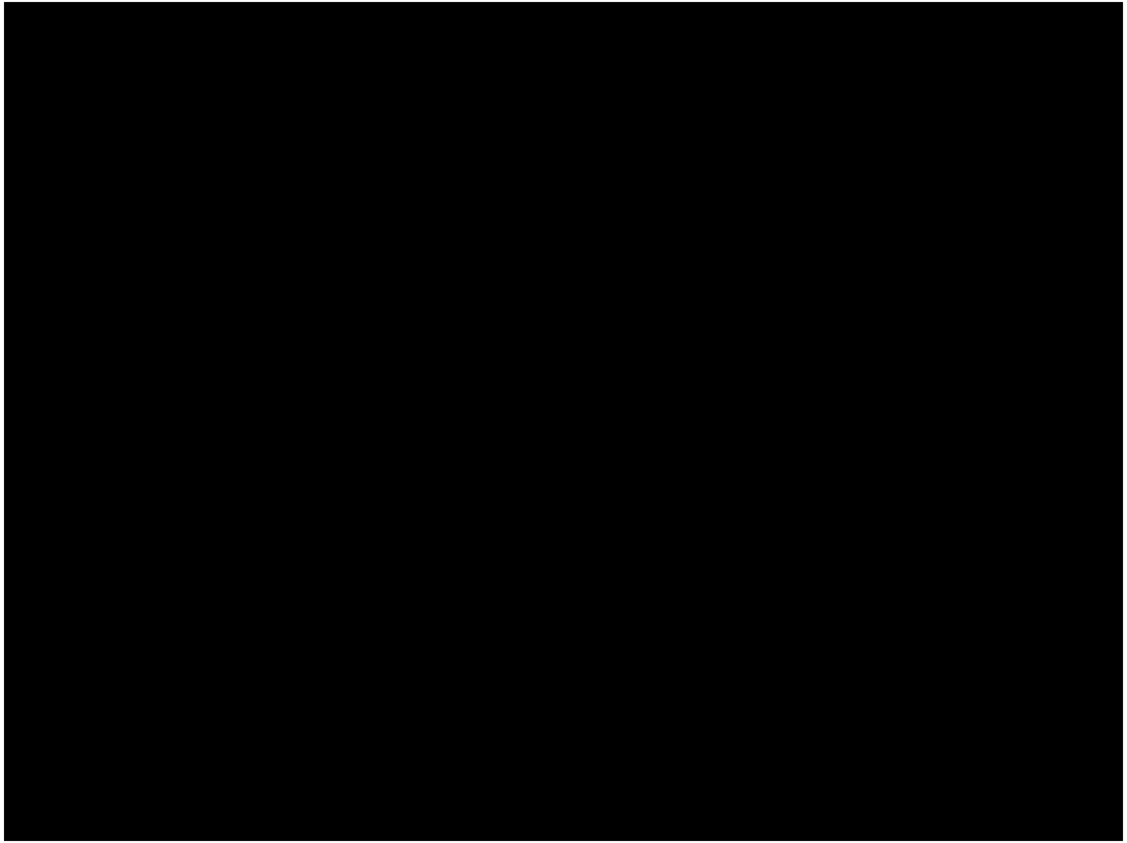
- R is Reward , D is discount Rate
- $R_{t=0} + R_{t=1}D + R_{t=2}D^2 + R_{t=3}D^3 + \ldots\ldots + R_{t=n}D^n$

Closer D is to 1, the more weight future rewards have. Closer to 0, future rewards don't count as much as immediate rewards

Before Training



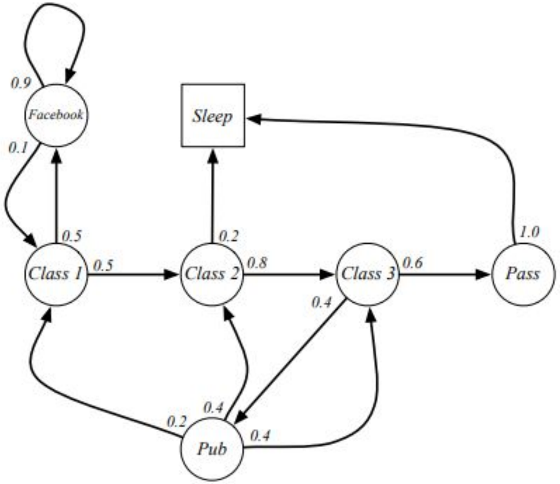
After Training



Reinforcement Learning

- A. Yang - Reinforcement Learning & Control
  - Definition of Markov Decision Processes
  - State, Action, Discount Factor, Possible Next States, and Reward Function
    - Adjust Certain Parameters based on what you are trying to achieve (Ex: Make Discount Factor = 0 to only focus on the immediate result of an action)
    - $$R(s_0, a_0) + \gamma R(s_1, a_1) + \gamma^2 R(s_2, a_2) + \cdots$$
- D. Silver (DeepMind) - Lecture: MDPs
  - Markov Processes, Markov Reward Processes, Markov Decision Processes
  - Class Example

Example: Student Markov Chain Episodes



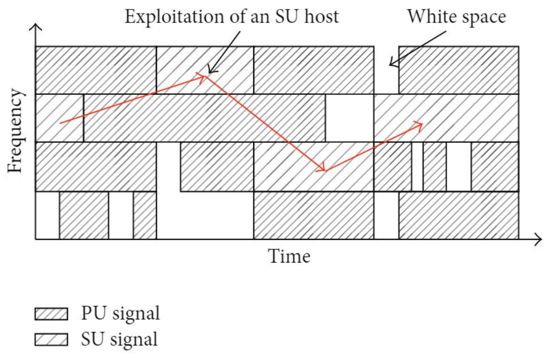
Sample episodes for Student Markov Chain starting from $S_1 = \text{C1}$

$$S_1, S_2, ..., S_T$$

- C1 C2 C3 Pass Sleep
- C1 FB FB C1 C2 Sleep
- C1 C2 C3 Pub C2 C3 Pass Sleep
- C1 FB FB C1 C2 C3 Pub C1 FB FB FB C1 C2 C3 Pub C2 Sleep

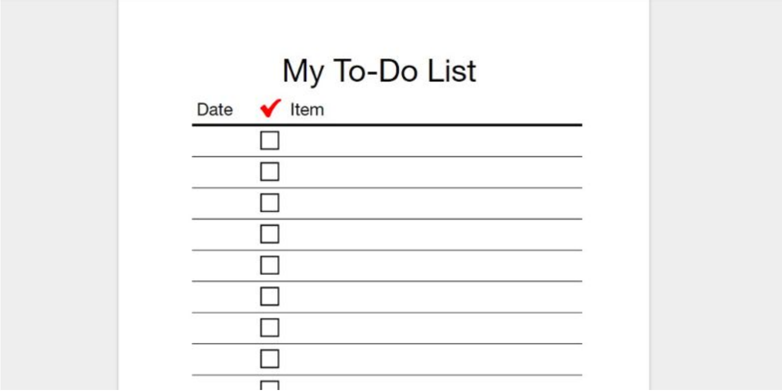
Yau et. al - Application of Reinforcement Learning in Cognitive Radio Networks:

- Q Learning and Greedy - How can I, the RL Model, efficiently find the episode with the most rewards for me?
- $$Q^i_{t+1}\left(s^i_t, a^i_t\right) \longleftarrow (1-\alpha)\, Q^i_t\left(s^i_t, a^i_t\right) + \alpha\left[r^i_{t+1}\left(s^i_{t+1}\right) + \gamma \max_{a \in A} Q^i_t\left(s^i_{t+1}, a\right)\right]$$
- 8 Areas of Research for RL
  - Dynamic Channel Selection
  - Channel Sensing



Todo

- Make list of reinforcement learning methods used on autonomous vehicles.
  - Compile list of compatible meta-learning methods.
- Be able to manipulate the different sensors on the CAT Vehicle in ROS and Gazebo
  - Receive input and output data from Gazebo
- Connect reinforcement learning to meta-learning.
- Determine a class of tasks we want to model.
- Explore past research in Dynamic Channel Selection/Channel Sensing
- Learn how to use TensorFlow to create a RL



Questions?