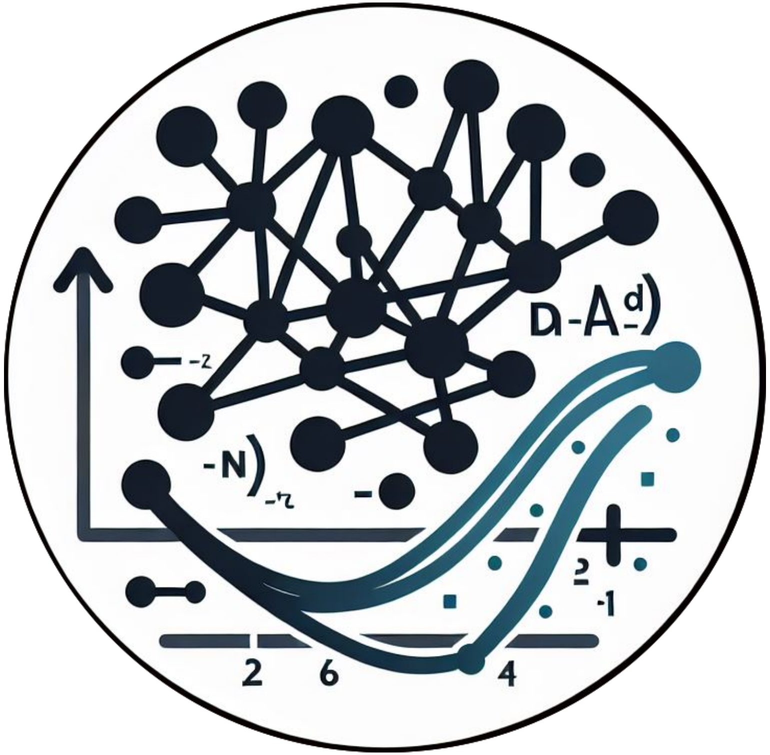
Deep Reinforcement Learning



Deep Learning for Engineers

Andrew Ning
aning@byu.edu

Motivation

Markov Decision Process

$$s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} s_3$$

$$s_{t+1} = f(s_t, a_t) \qquad s_{t+1} \sim P(s_{t+1} \mid s_t, a_t)$$

policy: $a_t = \mu(s_t) \qquad a_t \sim \pi(a_t \mid s_t)$

reward: $r(s_t, a_t)$

Trajectories (sequence of states)

$$\tau = (s_0, a_0, s_1, a_1, s_2, a_2 \cdots)$$

Reward

infinite-horizon

$$R(\tau) = \sum_{t=0}^{T} \gamma^t r_t = r_0 + \gamma r_1 + \gamma^2 r_2 + \dots$$

Expected Reward

$$J(\pi) = \mathop{\mathbb{E}}_{\tau \sim \pi}[R(\tau)] = \int P(\tau \mid \pi) R(\tau) d\tau$$

Value Functions

$$V^* = \max_a Q^*(s,a)$$

optimal policy

$$V^*(s) = \max_\pi \mathbb{E}[R(\tau) \mid s_0 = s]$$

expected reward if I start at state s & then follow optimal policy.

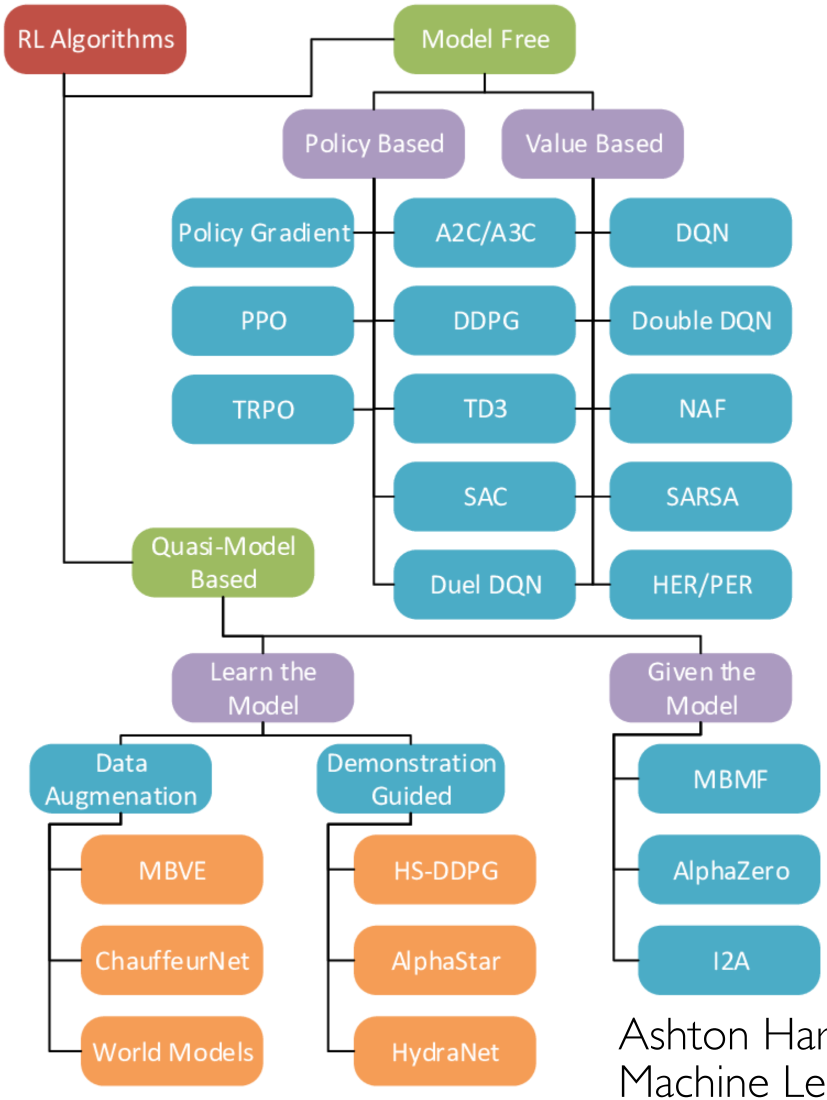
$$Q^*(s,a) = \max_\pi \mathbb{E}[R(\tau) \mid s_0 = s, a_0 = a]$$

expected reward if I start at s and take action a, and then follow optimal policy after that

Bellman Equations

$$V^*(s_t) = \max_{a_t} \mathbb{E}\left[r(s_t, a_t) + \gamma V^*(s_{t+1})\right]$$

$$Q^*(s_t, a_t) = \mathbb{E}\left[r(s_t, a_t) + \gamma \max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1})\right]$$

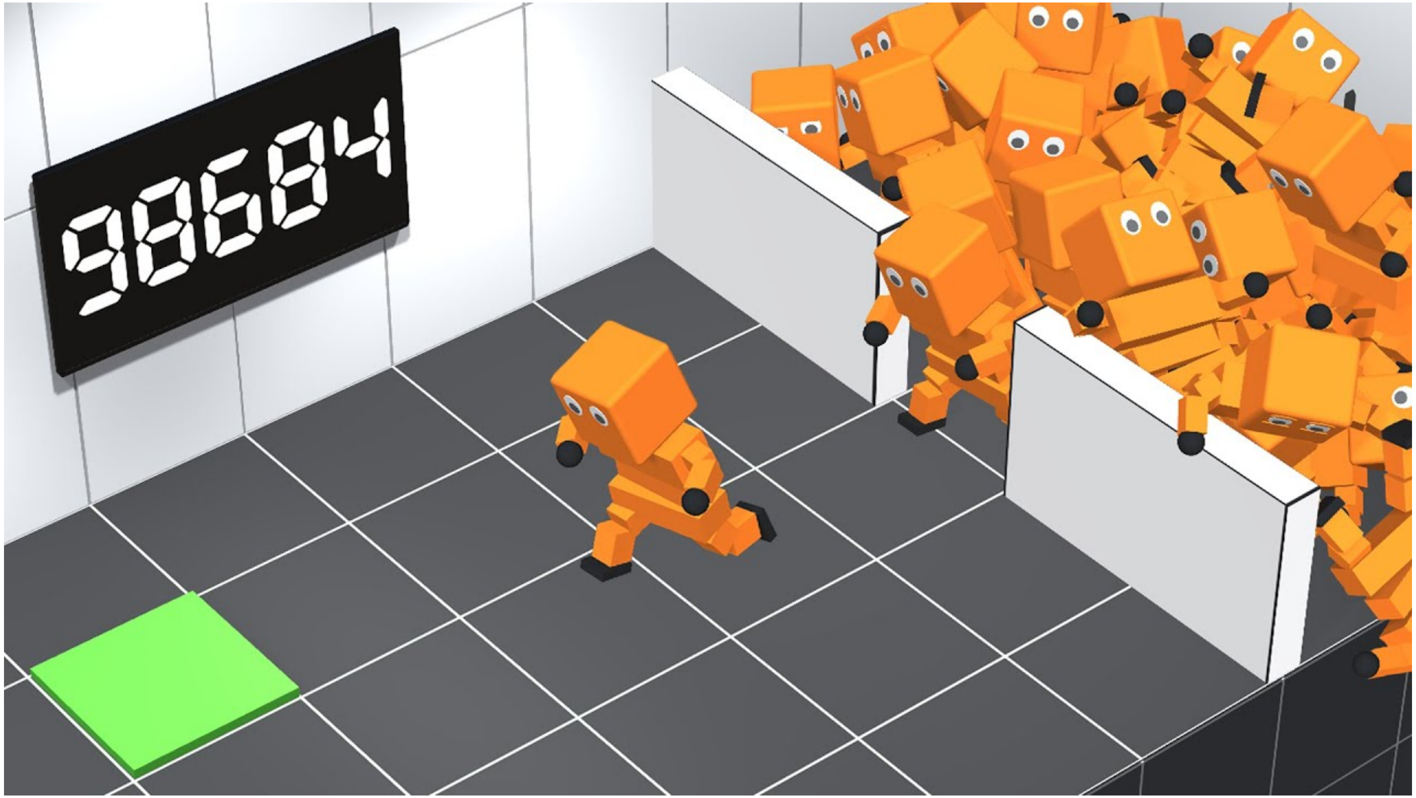


Ashton Harvey, Kathryn Blackmond Laskey, Kou-Chu Chang
Machine Learning Applications for Sensor Tasking with Non-Linear Filtering

Policy Gradient

Proximal Policy Optimization

PPO



Q-Learning


SUPER
MARIO WORLD™
© 1990, 1991 Nintendo

Soft Actor Critic