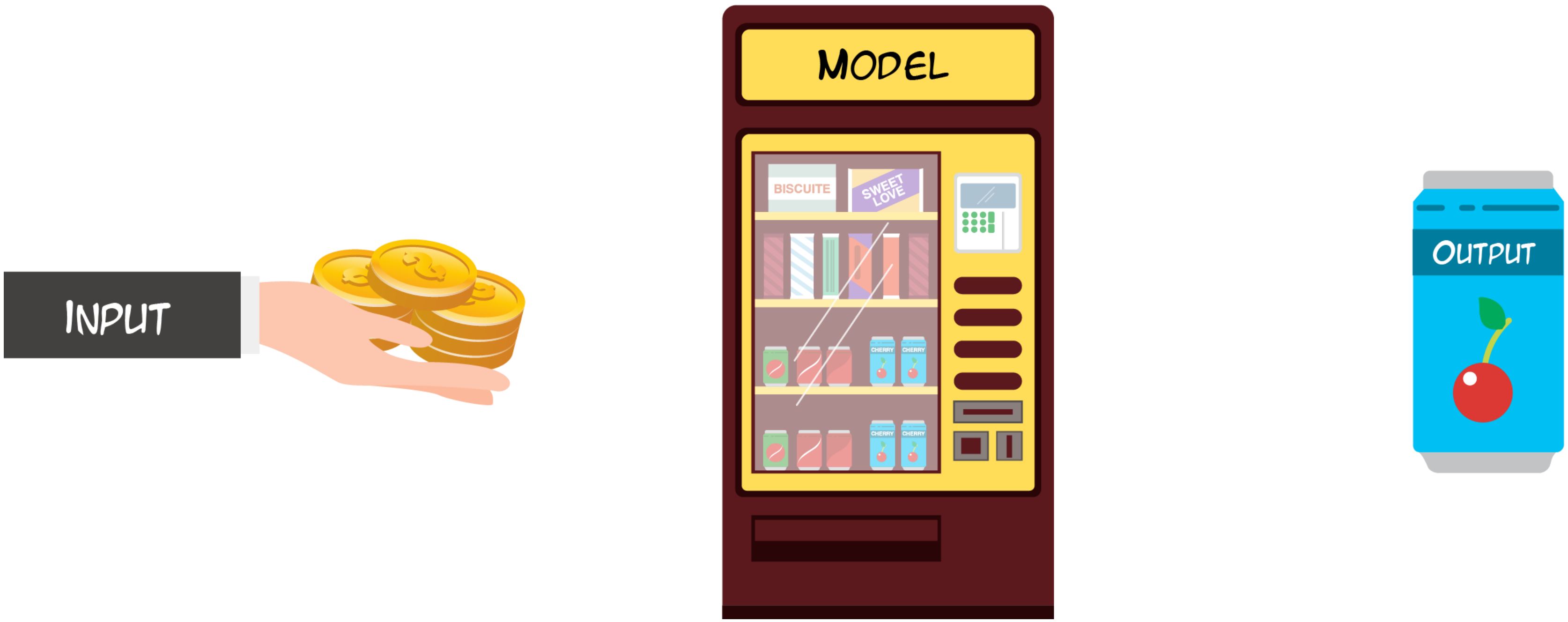
Reinforcement Learning

Jaley Dholakiya

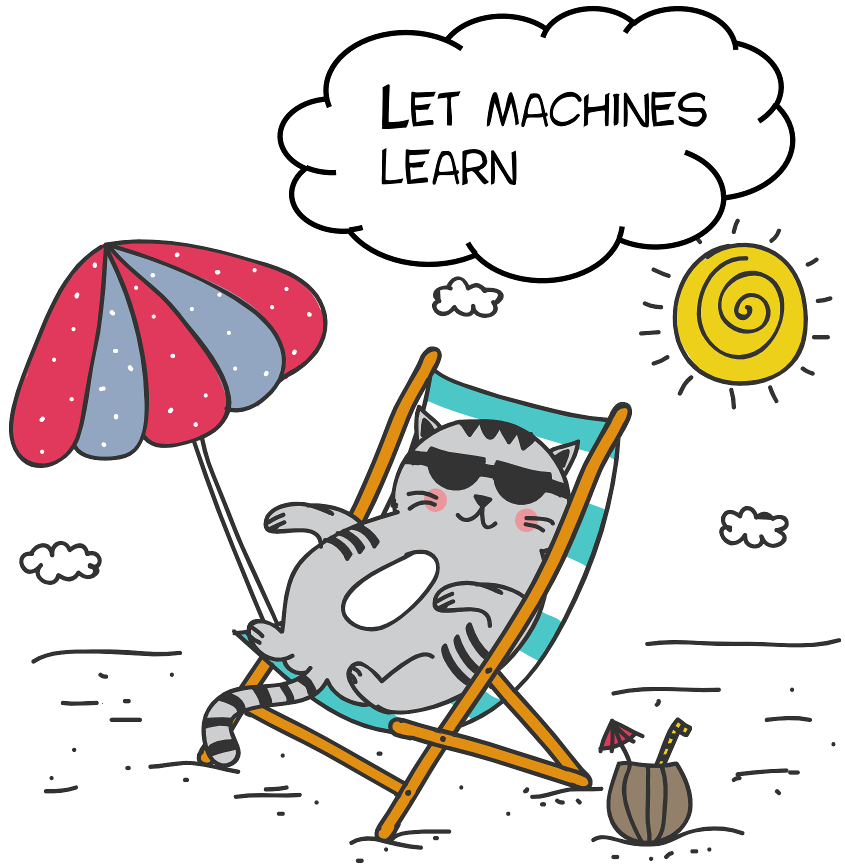
Recap : Deep Learning



$$y \approx f_w(x)$$

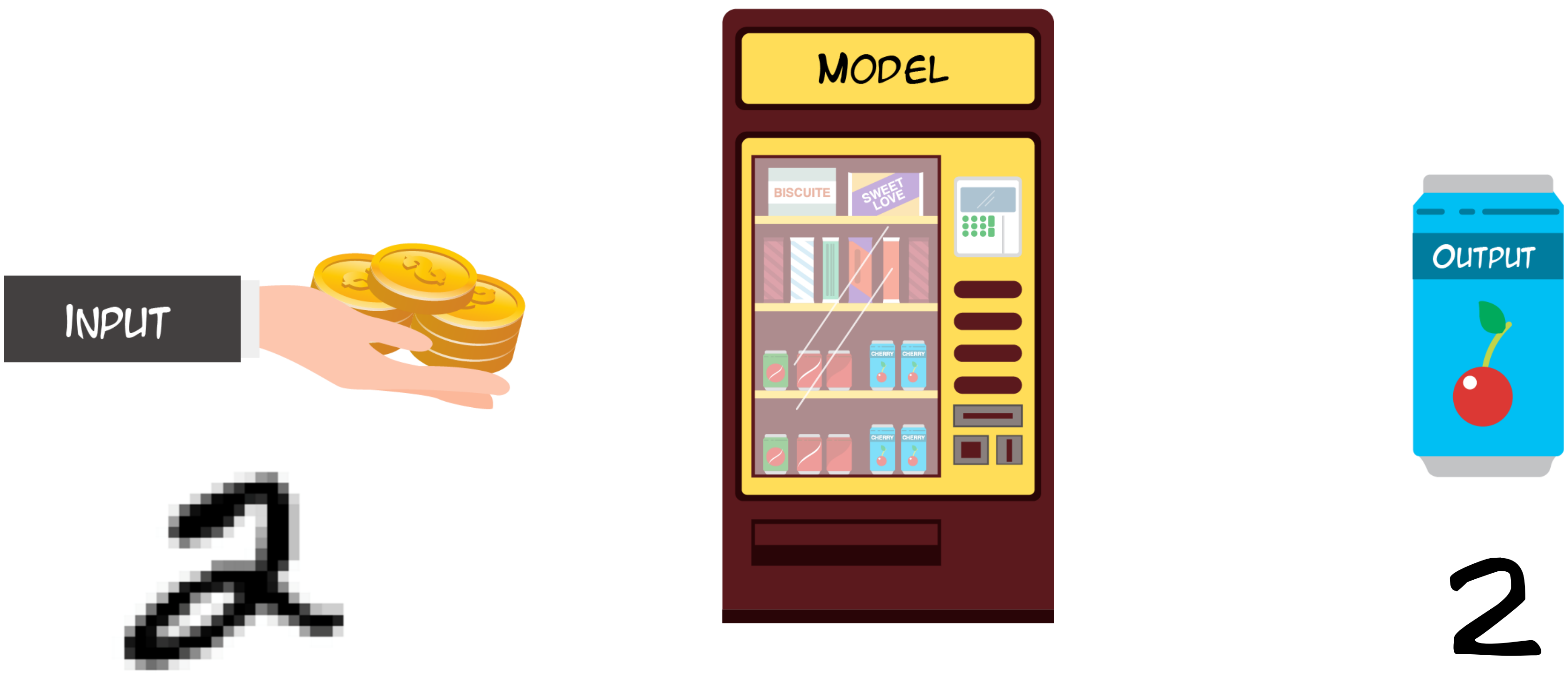
PROGRAMMING VS DEEP LEARNING

PROGRAMMING



DEEP LEARNING

MNIST Digit Classification

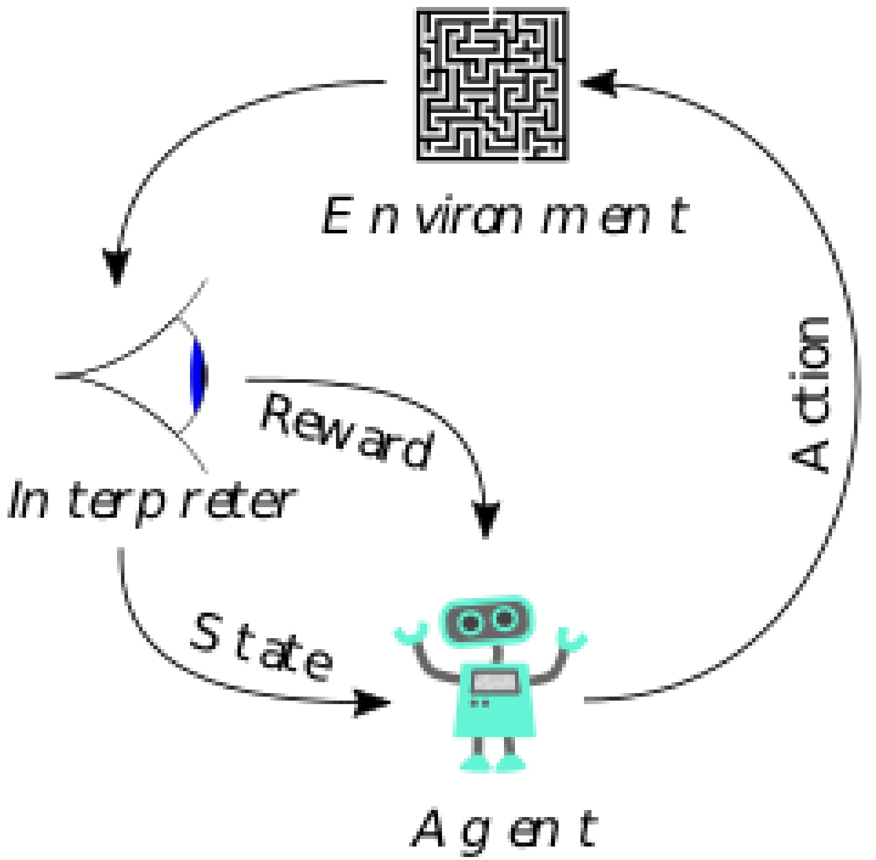


$$y \approx f_w(x)$$

Challenges in Deep learning

- Its Data Hungry
- No Active participation

Reinforcement Learning



RL vs DL

- Active Engagement
- Get rid of Manual Data Collection

Components of RL

Environment

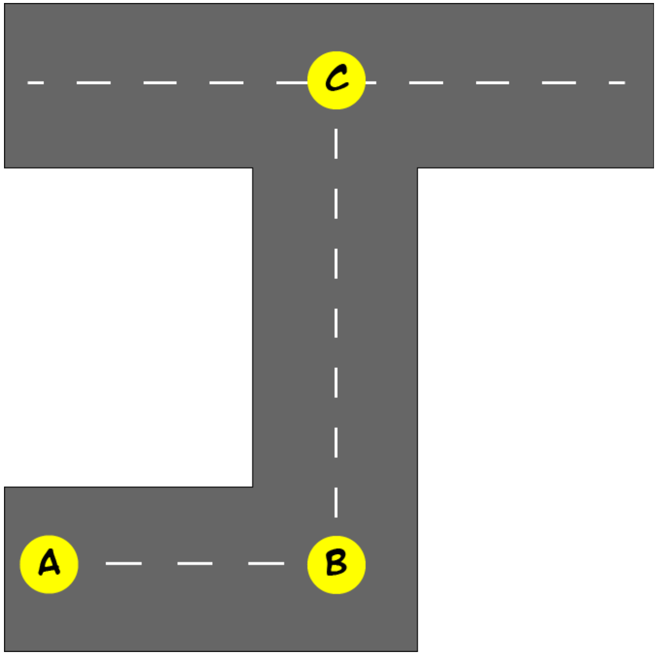
- Episodic vs Non-Episodic
- Fixed vs Dynamic
- Single vs Multiplayer

Video 1 : Components of RL

Gym Example

Goal of RL

State	Action
A	Go Straight
B	Take Left
C	Take Right or Left

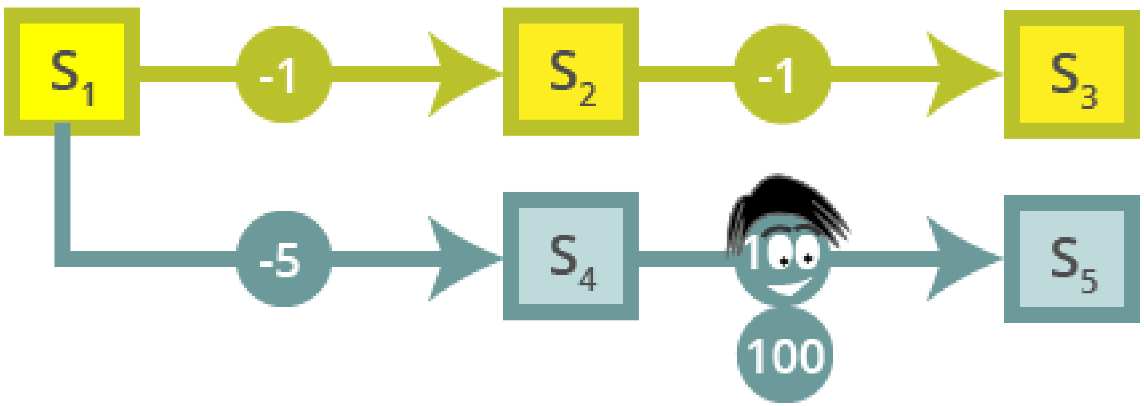


Which Action is best?

Most rewarding action

Immediate Reward?

Immediate Reward?



Discounted Reward

Types of RL Algorithms
(Based on Model)

- Model Dependent

  (Too much Math/ less useful)

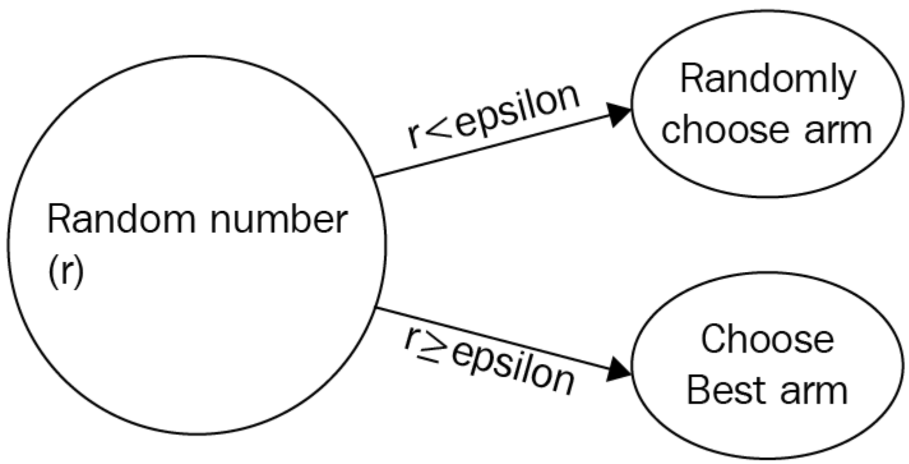
- Model Free

  (Less Math + More Fun)

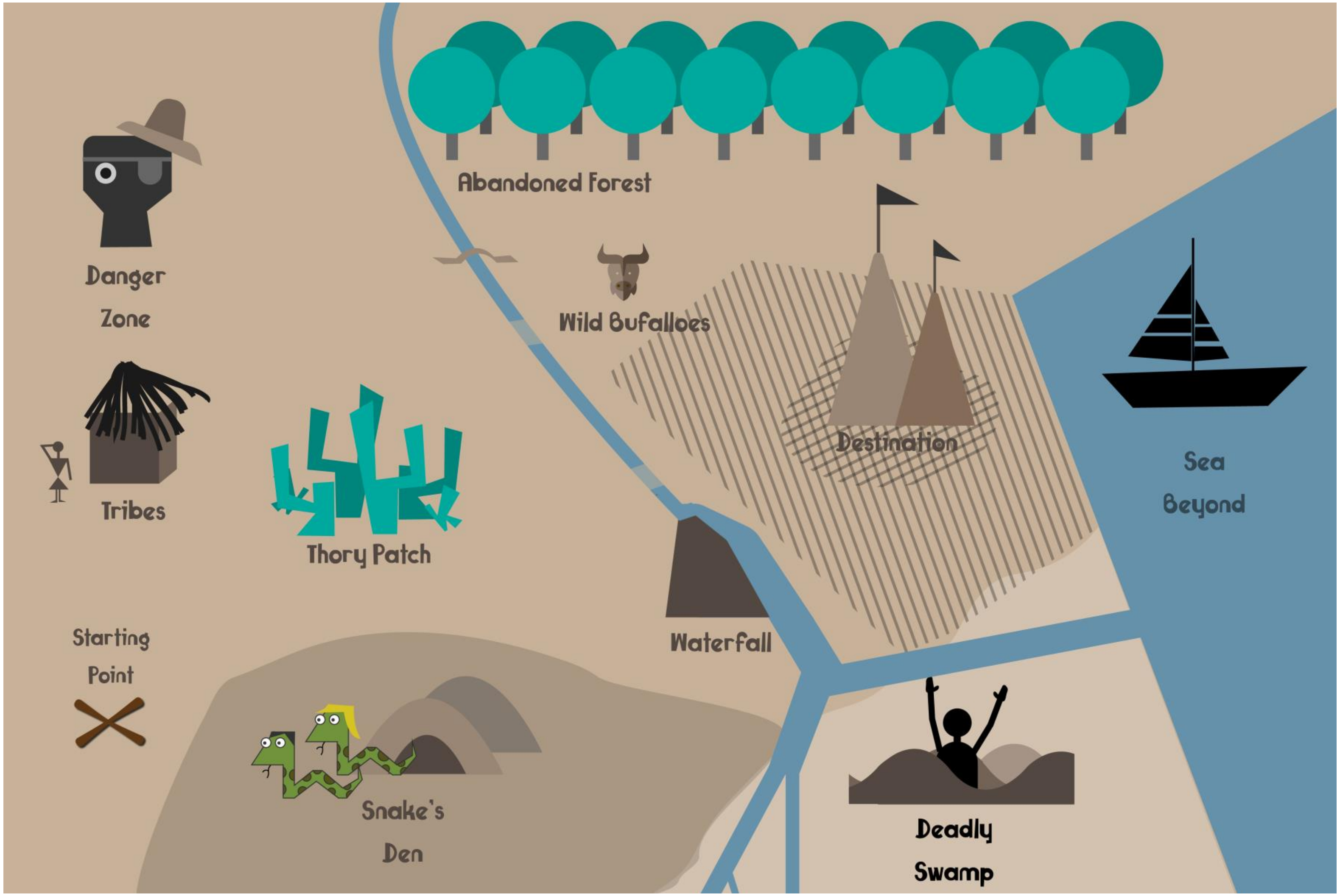
Types of RL Algorithms (Based on Policy)

- **Policy Evaluation**
  Estimate the state/action rewards for a given policy
- **Policy Control**
  Task is to **improve** a policy

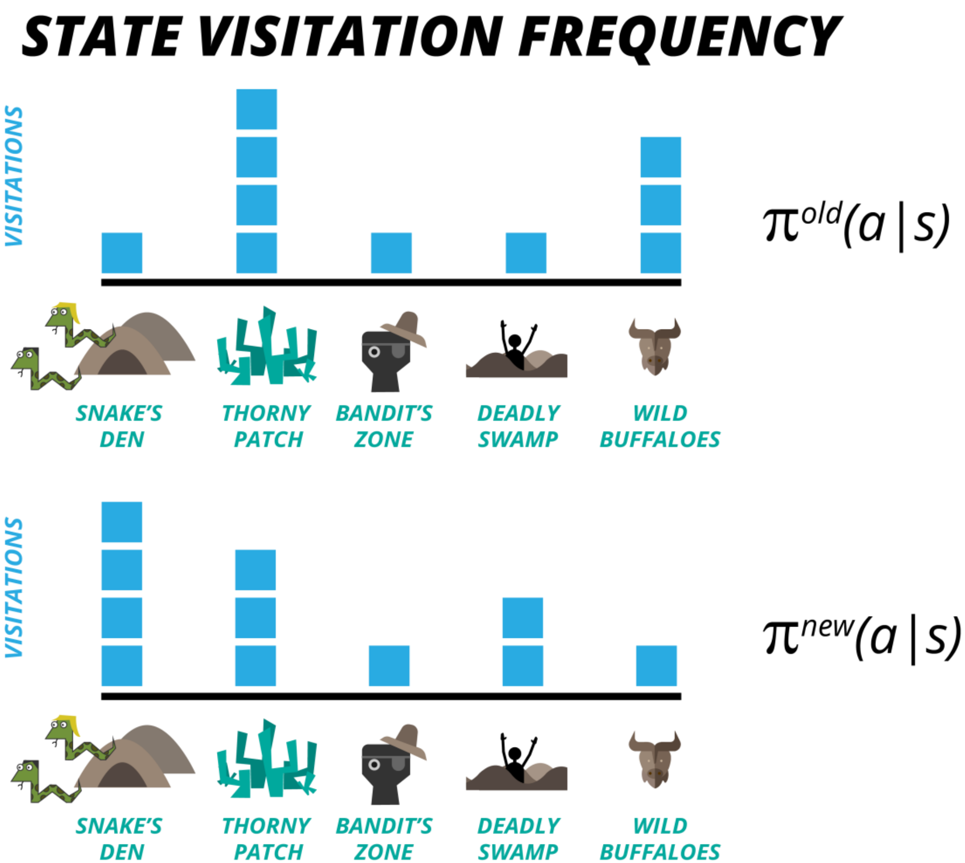
Policy Control
Epsilon greedy policy



Explore Exploit Problem



Explore Exploit Problem

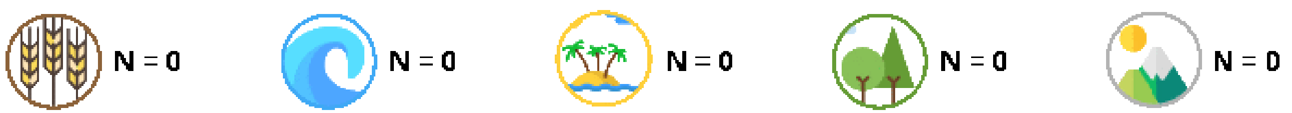


Monte Carlo Evaluation

Farm $\rightarrow$ Sea $\rightarrow$ Island $\rightarrow$ Forest $\rightarrow$ Dead

Sea $\rightarrow$ Island $\rightarrow$ Forest $\rightarrow$ Mountain $\rightarrow$ Forest $\rightarrow$ Dead

Forest $\rightarrow$ Sea $\rightarrow$ Island $\rightarrow$ Sea $\rightarrow$ Island $\rightarrow$ Forest $\rightarrow$ Dead



Monte Carlo Evaluation

Algorithm 1: Monte Carlo Evaluation

- $N(s) \leftarrow 0, \forall s \in S$
- $S(s) \leftarrow 0, \forall s \in S$
- **for** *each episode* **do**
  - **for** *each state(s)* $\in$ *episode* **do**
    - $G_t \leftarrow 0$
    - $i = 0$
    - **for** $t \leftarrow [t_{curr}, T]$ **do**
      - $G_t \leftarrow G_t + \gamma^i R_t$
      - $i \leftarrow i + 1$
    - $N(s) \leftarrow N(s) + 1$
    - $S(s) \leftarrow S(s) + G_t$
    - $V(s) \leftarrow S(s)/N(s)$

Algorithm 2: Monte Carlo Evaluation with incremental updates

- $N(s) \leftarrow 0, \forall s \in S$
- $S(s) \leftarrow 0, \forall s \in S$
- **for** *each episode* **do**
  - **for** *each state(s)* $\in$ *episode* **do**
    - $G_t \leftarrow 0$
    - $i = 0$
    - **for** $t \leftarrow [t_{curr}, T]$ **do**
      - $G_t \leftarrow G_t + \gamma^i R_t$
      - $i \leftarrow i + 1$
    - $V(s) \leftarrow V(s) + \alpha(G_t - V(s))$
    - *or*
    - $N(s) \leftarrow N(s) + 1$
    - $V(s) \leftarrow V(s) + \frac{1}{N(s)}(G_t - V(s))$

Temporal Difference

$$G_t(s_t) = R_{t+1} + \gamma R_{t+2} + \gamma^{T-2} \cdots R_{T-1}$$

We can write reward for next state as

$$G_{t+1}(s_{t+1}) = R_{t+2} + \gamma R_{t+3} + \gamma^{T-3} \cdots R_{T-1}$$

Therefore we can also write the first equation as

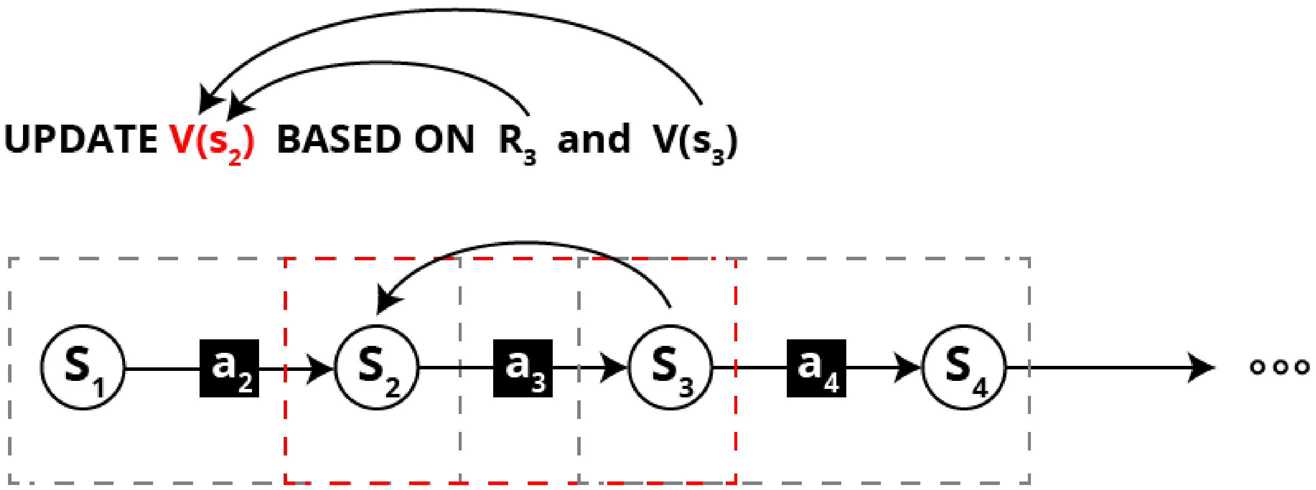
$$G_t(s_t) = R_{t+1} + \gamma \overbrace{\{R_{t+2} + \gamma^{T-3} \cdots R_{T-1}\}}^{G_{t+1}(s_{t+1})}$$

and we all know we want to update value function, closer to discounted reward $(G \leftarrow V)$, therefore our new update for TD Learning becomes

$$V(s_t) = V(s) + \alpha(\hat{G}_t - V(s))$$

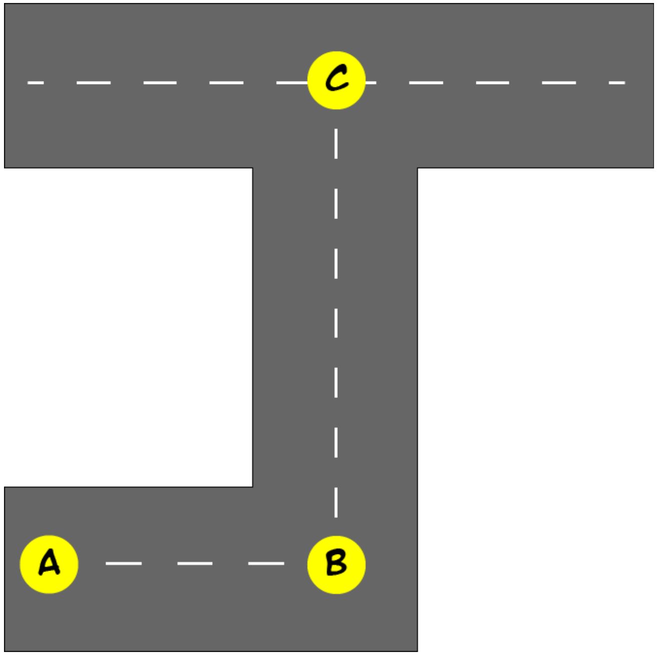
$$V(s_t) = V(s_t) + \alpha(R_{t+1} + \gamma V(s_{t+1}) - V(s_t))$$

Temporal Difference



Reminder : Goal of RL

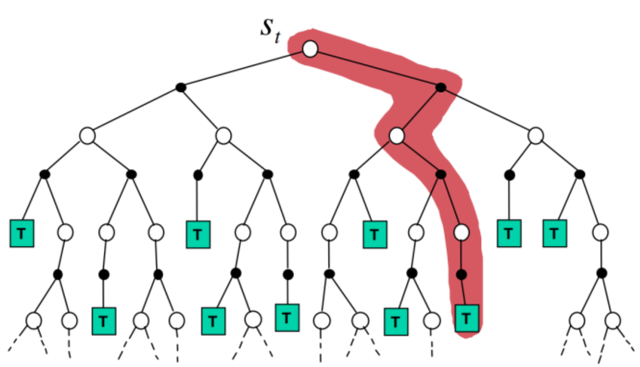
State	Action
A	Go Straight
B	Take Left
C	Take Right or Left



MC vs TD

Monte Carlo

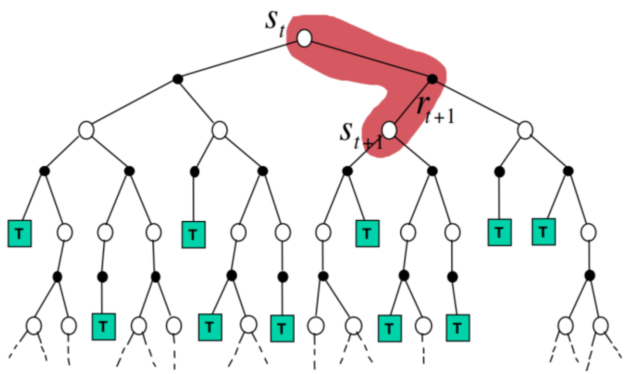
$V(S_t) \leftarrow V(S_t) + \alpha\,(G_t - V(S_t))$



$$q(a_t|s_t) = q(a_t|s_t) + \alpha(G_t - q(a_t|s_t))$$

Temporal Difference

$V(S_t) \leftarrow V(S_t) + \alpha\,(R_{t+1} + \gamma V(S_{t+1}) - V(S_t))$



$$q(a_t|s_t) = q(a_t|s_t) + \alpha((R_t + \gamma q(a_{t+1}|s_{t+1})) - q(a_t|s_t))$$

TD Control + Greedy = SARSA
(State Action Reward State Action)

Policy Evaluation

$$q(a_t|s_t) = q(a_t|s_t) + \alpha((R_t + \gamma q(a_{t+1}|s_{t+1})) - q(a_t|s_t))$$

$$q(a_t|s_t) = \boldsymbol{(1-\alpha)}q(a_t|s_t) + \boldsymbol{\alpha}(R_t + \gamma q(a_{t+1}|s_{t+1}))$$

Policy Update

$\epsilon - greedy\ approach$

SARSA

$$q(a_t|s_t) = \boldsymbol{(1-\alpha)}q(a_t|s_t) + \boldsymbol{\alpha}(R_t + \gamma q(a_{t+1}|s_{t+1}))$$

Q Learning

Requires only SARS pairs

$$q(a_t|s_t) = \boldsymbol{(1-\alpha)}q(a_t|s_t) + \boldsymbol{\alpha} * max(R_t + \gamma q(a_*|s_{t+1}))$$

Demo Time

(GRID-WORLD)

Goal of Q Learning

- Estimating Action Value Function (Q)
- Update Q function via moving avg

$$q(a_t|s_t) = (\mathbf{1} - \boldsymbol{\alpha})q(a_t|s_t) + \boldsymbol{\alpha} * max(R_t + \gamma q(a_*|s_{t+1}))$$

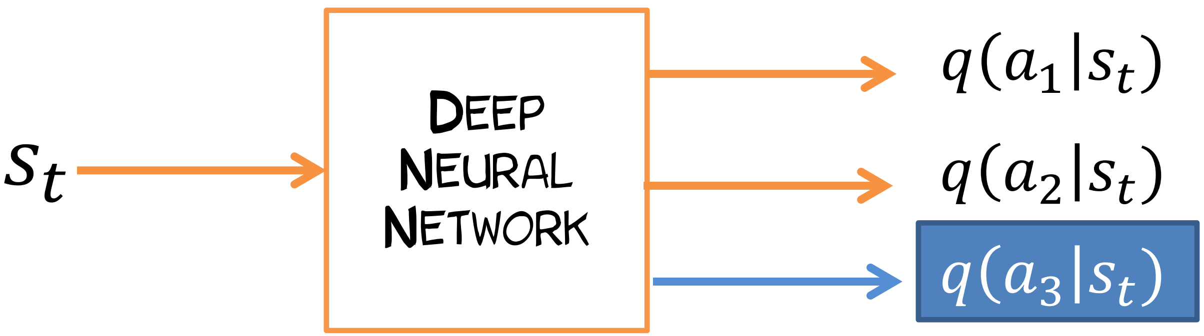
Problem with Q Learning

- Can't handle near Infinite states (images)
- Can't handle State as feature vector

Q Learning

$$q(a_t|s_t) = \color{red}{(\mathbf{1} - \boldsymbol{\alpha})}q(a_t|s_t) + \color{red}{\boldsymbol{\alpha} * max}(R_t + \gamma q(\color{red}{a_*}|s_{t+1}))$$

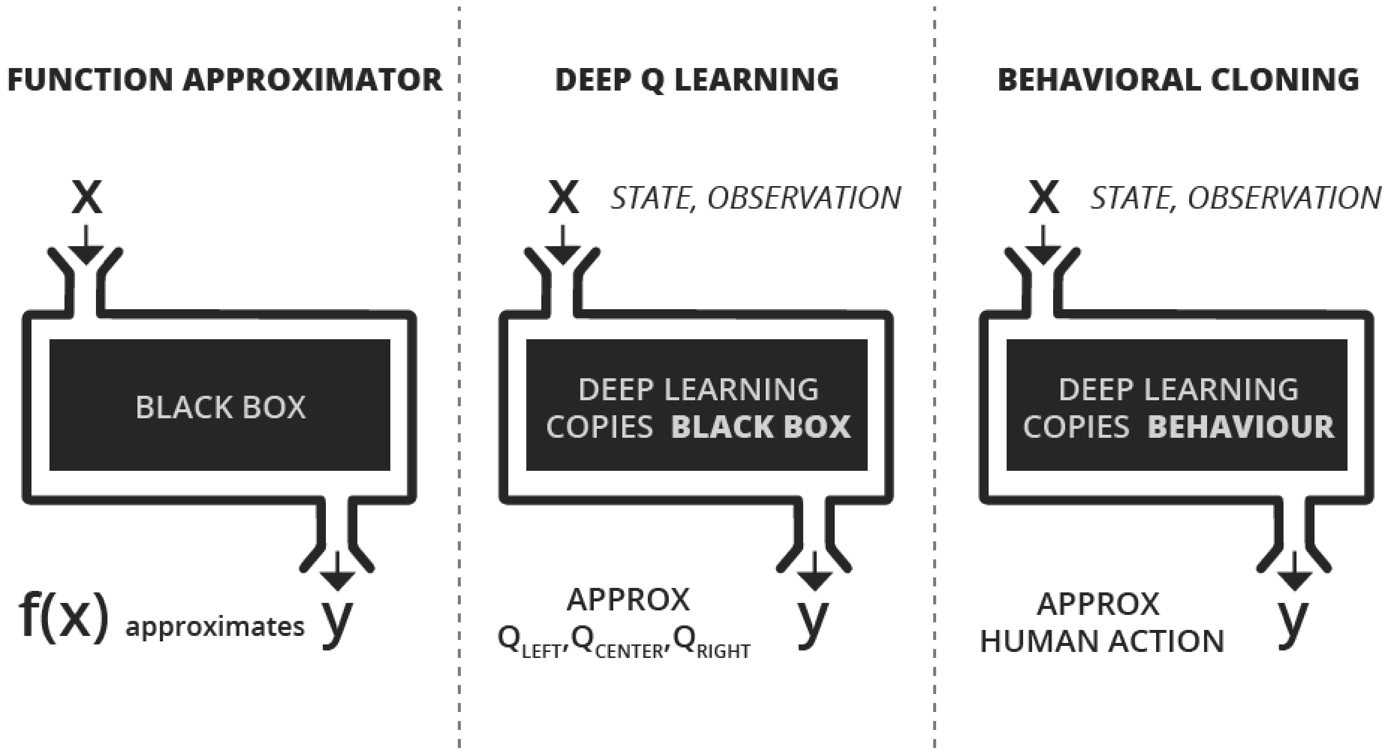
Deep Q Learning



Replace chosen action's value with $\color{red}{max}(R_t + \gamma q(\color{red}{a_*}|s_{t+1}))$

Update neural network by setting label for $q(a_t|s_t)\ as$ $\color{red}{max}(R_t + \gamma q(\color{red}{a_*}|s_{t+1}))$

$$q(a_t|s_t) = neural - network(s_t)$$

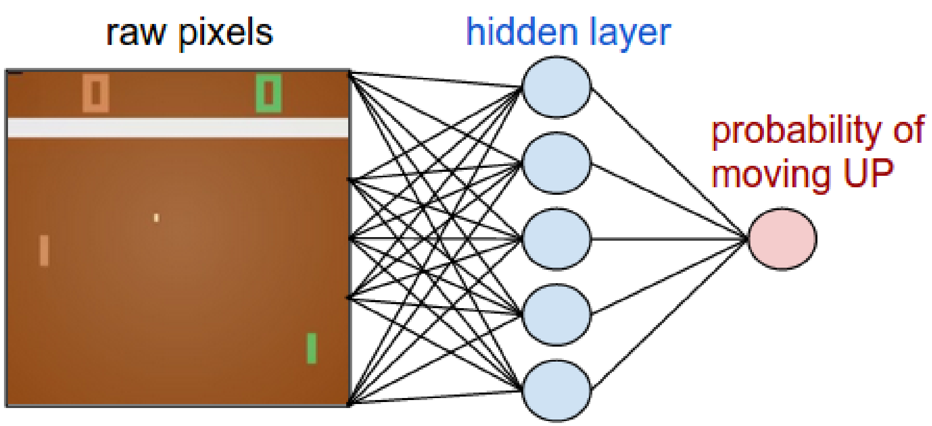

FUNCTION APPROXIMATOR
x
BLACK BOX
f(x) approximates y
DEEP Q LEARNING
x STATE, OBSERVATION
DEEP LEARNING COPIES BLACK BOX
APPROX $Q_{LEFT}, Q_{CENTER}, Q_{RIGHT}$
y
BEHAVIORAL CLONING
x STATE, OBSERVATION
DEEP LEARNING COPIES BEHAVIOUR
APPROX HUMAN ACTION
y

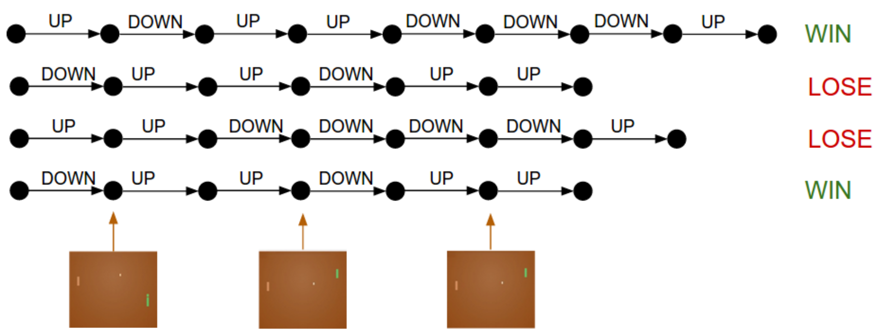
DEMO TIME

(CARTPOLE IN OPENAI-GYM)

Policy Gradient Approaches

Change policy rather than Q values





POLICY ENTROPY =0.9

POLICY ENTROPY =0.1

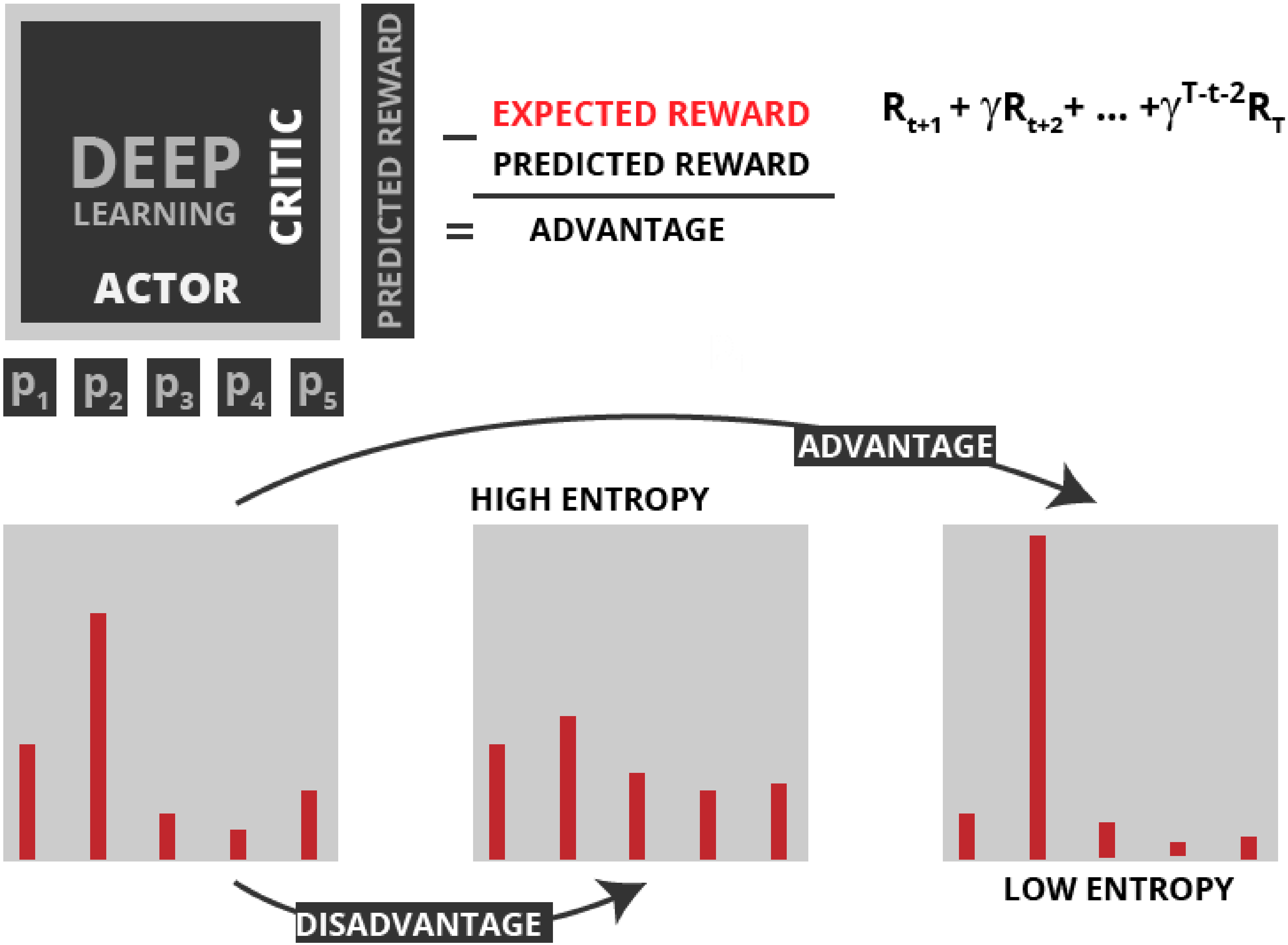
Actor Critic Model

CRITIC

Understands each shot(State), based on rewards

ACTOR

Learns to act based on ***advantage factor*** *given by critic.*

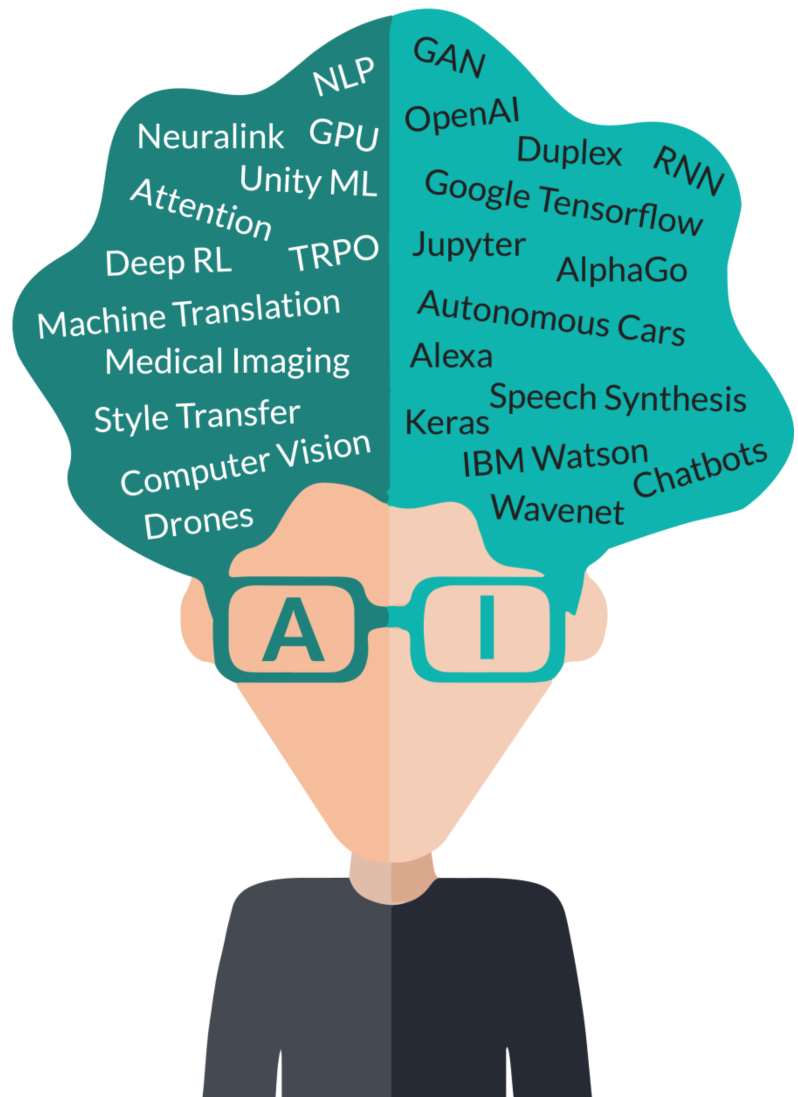

DEEP
LEARNING
ACTOR
CRITIC
PREDICTED REWARD
EXPECTED REWARD
PREDICTED REWARD
ADVANTAGE
$R_{t+1} + \gamma R_{t+2} + \ldots + \gamma^{T-t-2} R_T$
p_1 p_2 p_3 p_4 p_5
ADVANTAGE
HIGH ENTROPY
DISADVANTAGE
LOW ENTROPY

Other cool stuff

Proximal policy optimization
Monte Carlo Tree Search

Subscribe

www.youtube.com/c/crazymuse



Contact Details

jaley.dholakiya@crazymuse.in
8105207901
Github link : www.github.com/crazymuse
Medium Link : www.medium.com/crazymuse

Ping me for any doubt