Deep Reinforcement Learning

<Hetav Pandya/>

Introduction

{

"Name" : "Hetav Pandya - HP",

“Work" : “Software Engineer",

“Affiliation" : [“Arista Networks”, “UBC AI”],

“Education" : "University of Toronto",

"Experience" : "Intel Corporation, Bell, General Motors"

}

Agenda

{

"ItemsForToday" : [

"Introduction to Deep RL",

"What are Policy Gradient Methods?",

"What are DQNs?",

"Demo with a real-life application",

"Q&A a.k.a Roasting Session"]

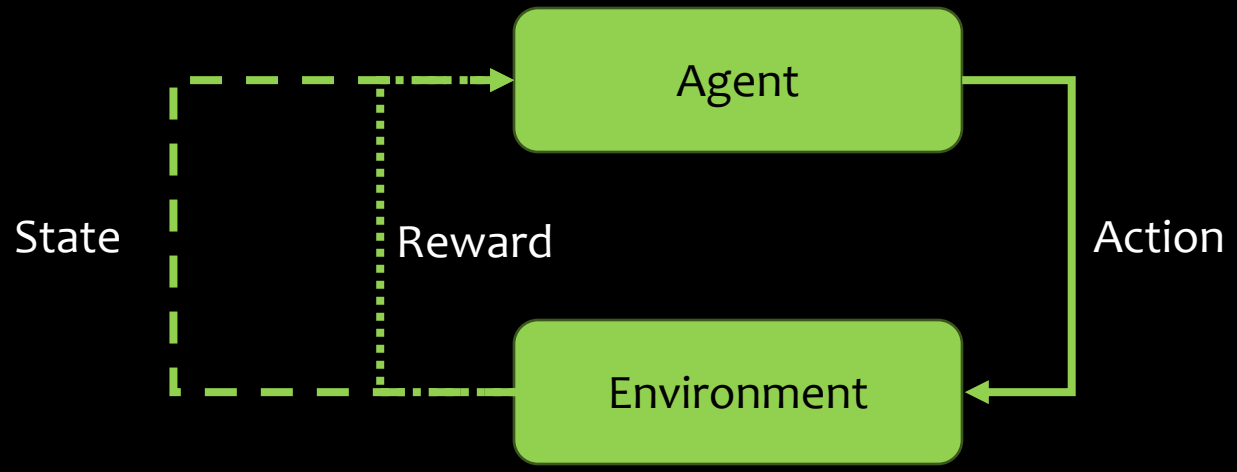
}

```
targetAudience
{
        “Audience" : [
                “Have a calculus background”,
                “Have a probability background”,
                “Want to use Deep RL in your work”,
                “Are comfortable using Python”,
                ]
}
```

What is Reinforcement Learning?

“Approach for learning decision making and control from experience”

– CS285 UC Berkeley

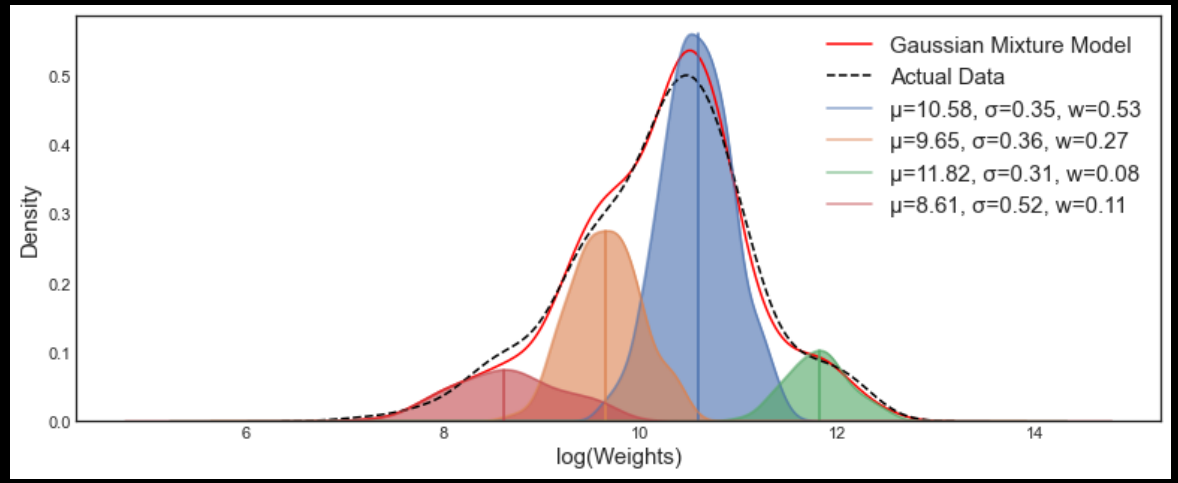


What is DEEP Reinforcement Learning

Historically we used to use approximators like gaussian distributions, linear approximators, sinusoidal regressors…

Now we use ‘Deep’ Neural Networks.

Why?



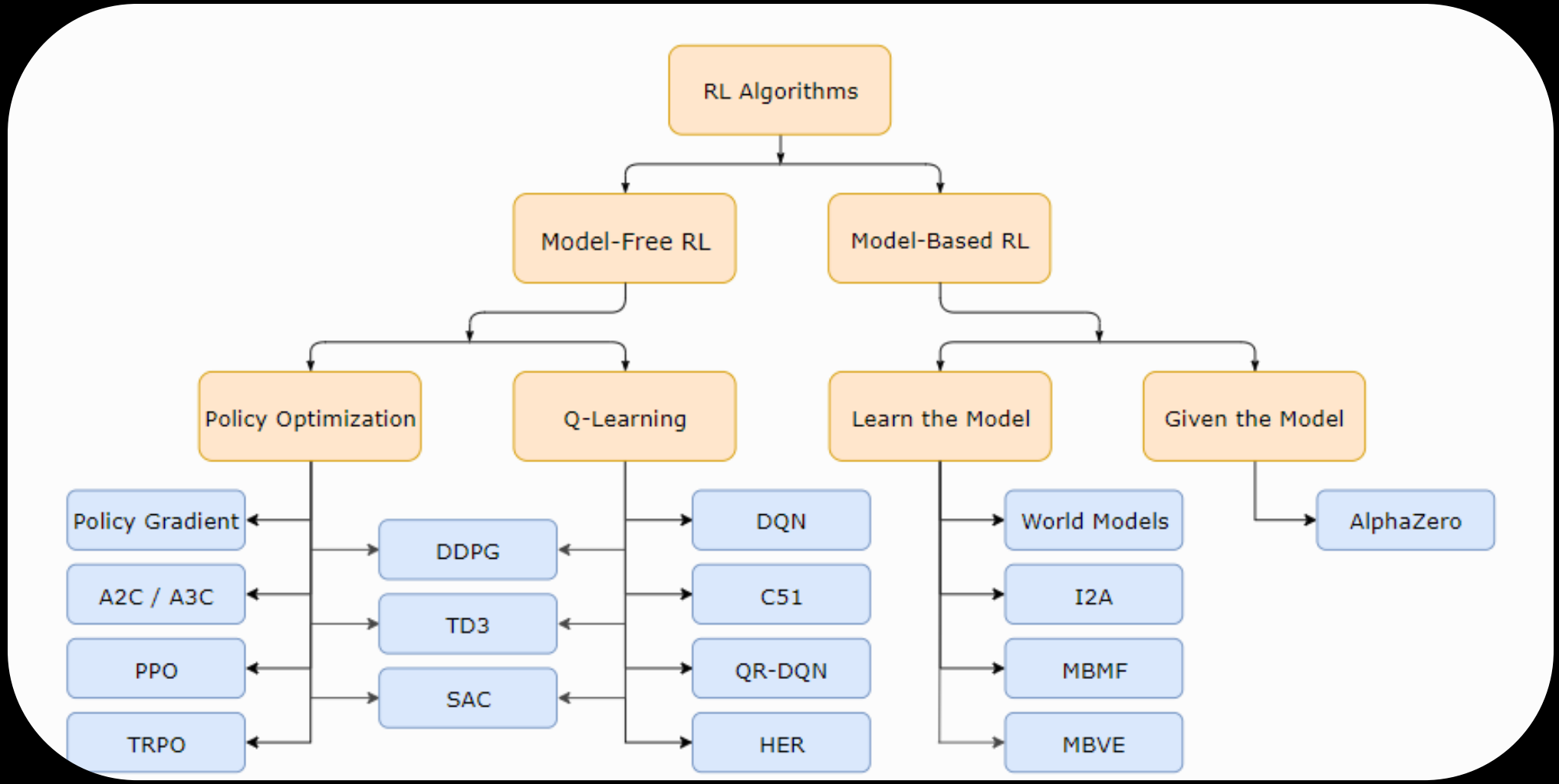
Where do we use RL?

It is not just games and robots!

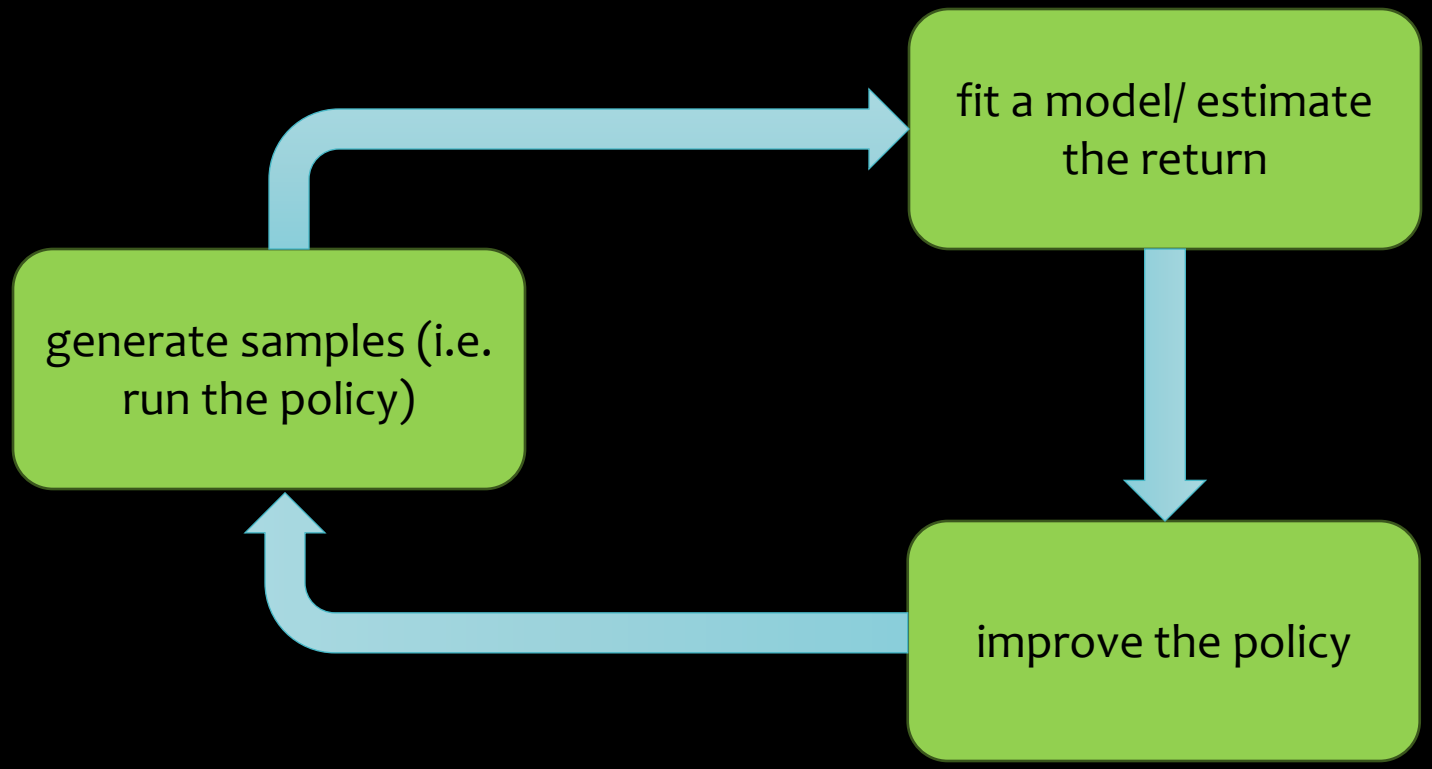
- For chip design[1]
- For improving language models[2]
- For inventory management
- For image generation [3]

[1] https://ai.googleblog.com/2020/04/chip-design-with-deep-reinforcement.html
[2] https://huggingface.co/blog/rlhf
[3] Kevin Black*, Michael Janner*, Yilun Du, Ilya Kostrikov, Sergey Levine. Training Diffusion Models with Reinforcement Learning

Types of RL Algorithms

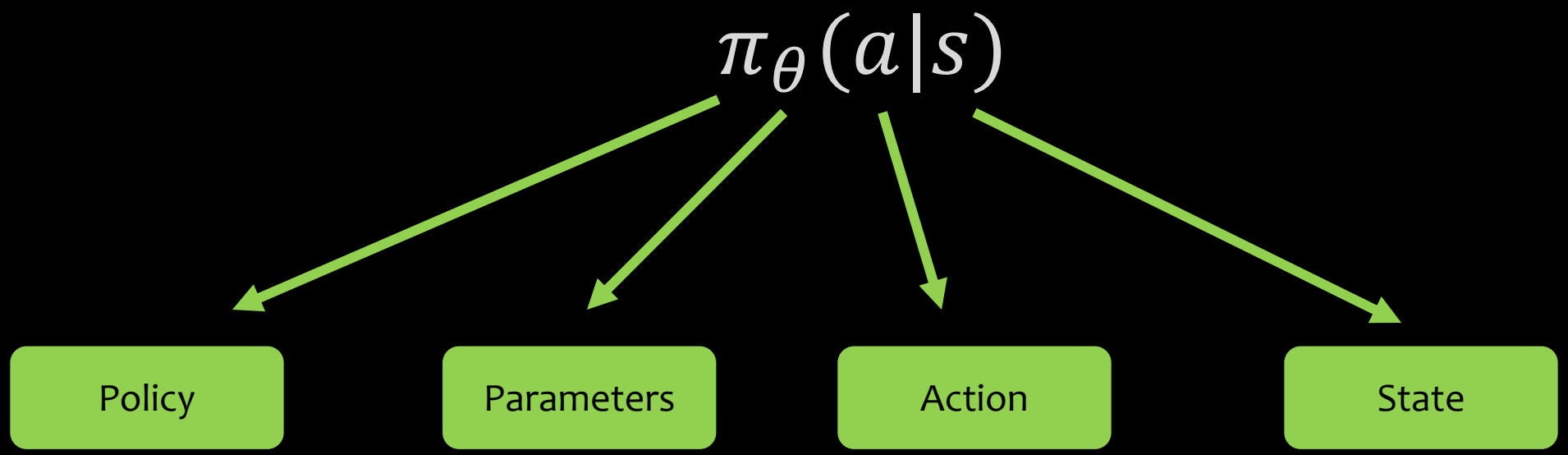


The trick to understand any RL algorithm

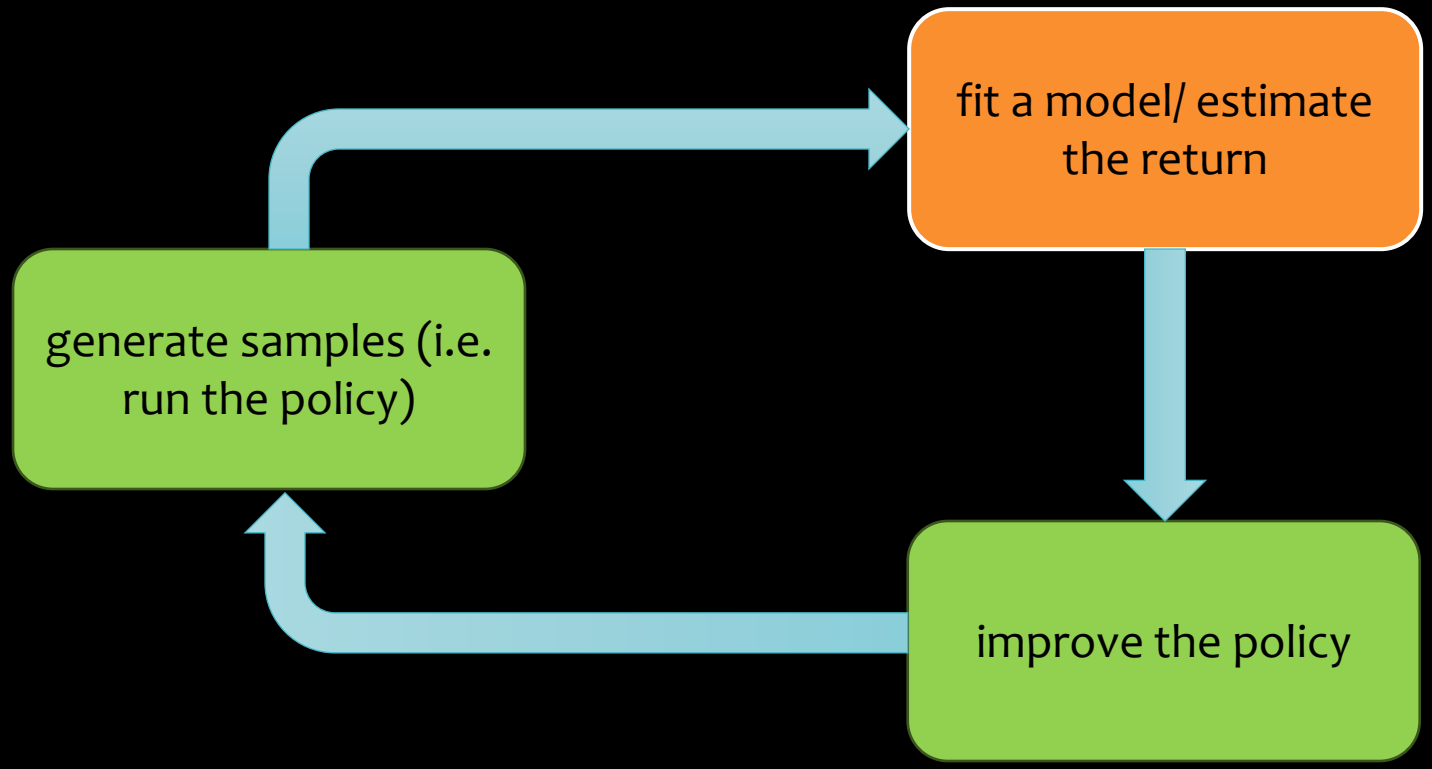


Source: UC Berkeley CS 285 – Lecture 4 Introduction to Reinforcement Learning

Policy Gradient / REINFORCE Algorithm



The trick to understand any RL algorithm



Source: UC Berkeley CS 285 – Lecture 4 Introduction to Reinforcement Learning

The goal of a policy gradient algorithm

$$J(\theta) = E_{T \sim \pi_\theta}\left[\sum_{t=0}^{T} R(s_t, a_t)\right]$$

- Total reward
- Expectation over the current policy
- Add all the rewards over T steps
- The reward depends on the state and action pair

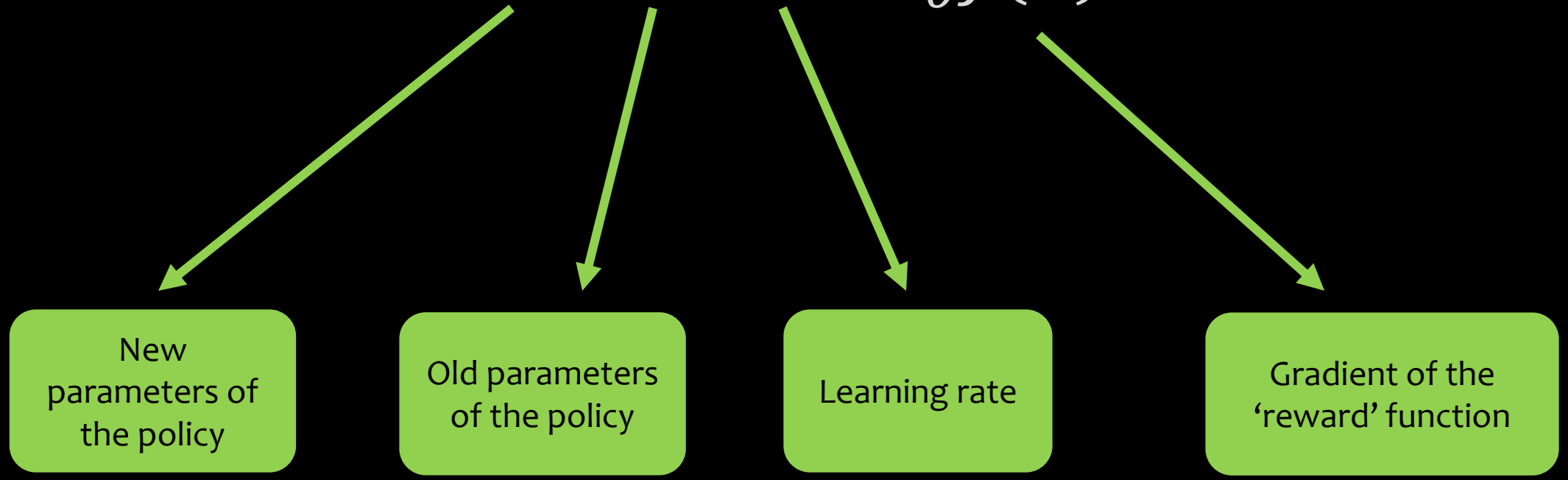
How to improve the policy ‘gradient’ algorithm?

$$\nabla_\theta J(\theta) = E_{T \sim \pi_\theta}\left[\sum_{t=0}^{T} \nabla_\theta \log(\pi_\theta(a_t|s_t))\, R_t\right]$$

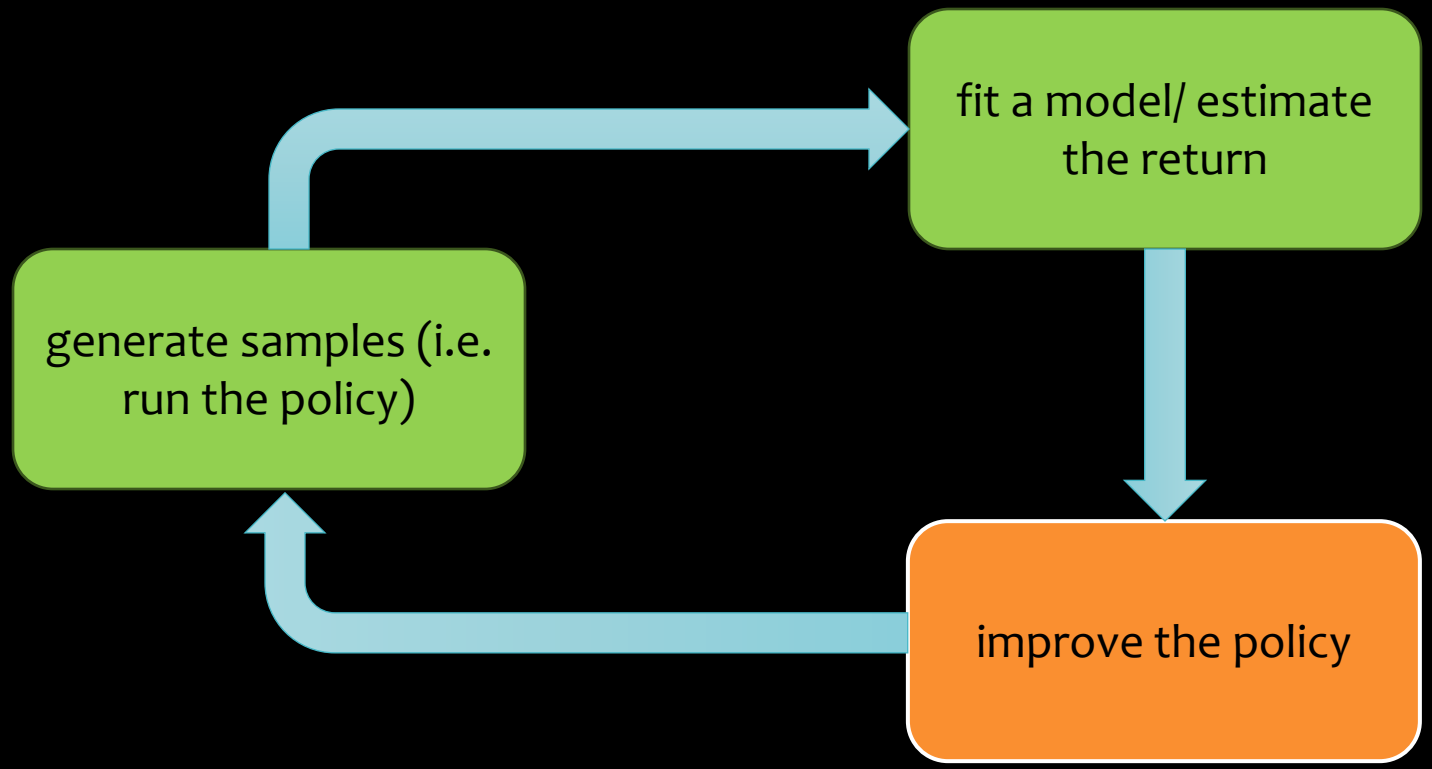
- Gradient of total reward
- Expectation over the current policy
- Summation over T steps
- Magic a.k.a complex math

How to improve the policy ‘gradient’ algorithm?

$$\theta \leftarrow \theta + \alpha * \nabla_\theta J(\theta)$$



The trick to understand any RL algorithm



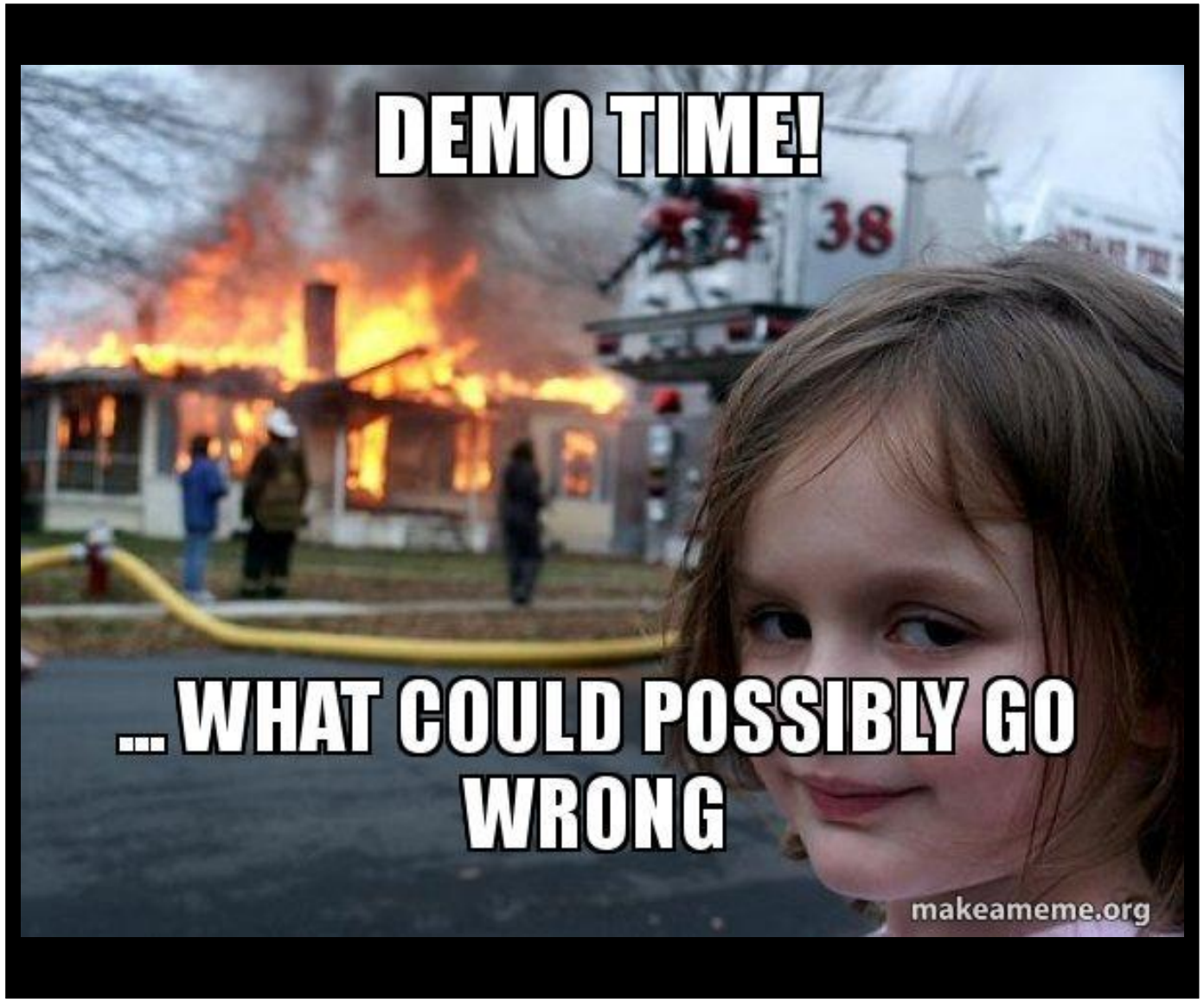
Source: UC Berkeley CS 285 – Lecture 4 Introduction to Reinforcement Learning

Practical implementation

- The math is not necessary ☺
- Has prebuilt optimizations
  - Variance reduction
  - Leveraging causality theorems
- Libraries like PyTorch have AutoGrad support

Algorithm -> Business Value

- Enhancing recommendation engines to provide more personalized content
- Optimizing advertising strategies, such as prices in real-time to maximize conversion rates
- Implementing adaptive pricing strategies that adjust prices based on demand
- Improving inventory management and logistics by dynamically adjusting order quantities



Demo

Making an inventory management system using REINFORCE (policy gradient) algorithm

The Deep Q-Network Algorithm

It uses **Deep** Neural **Networks** to learn **Q** values…

What are Q values?

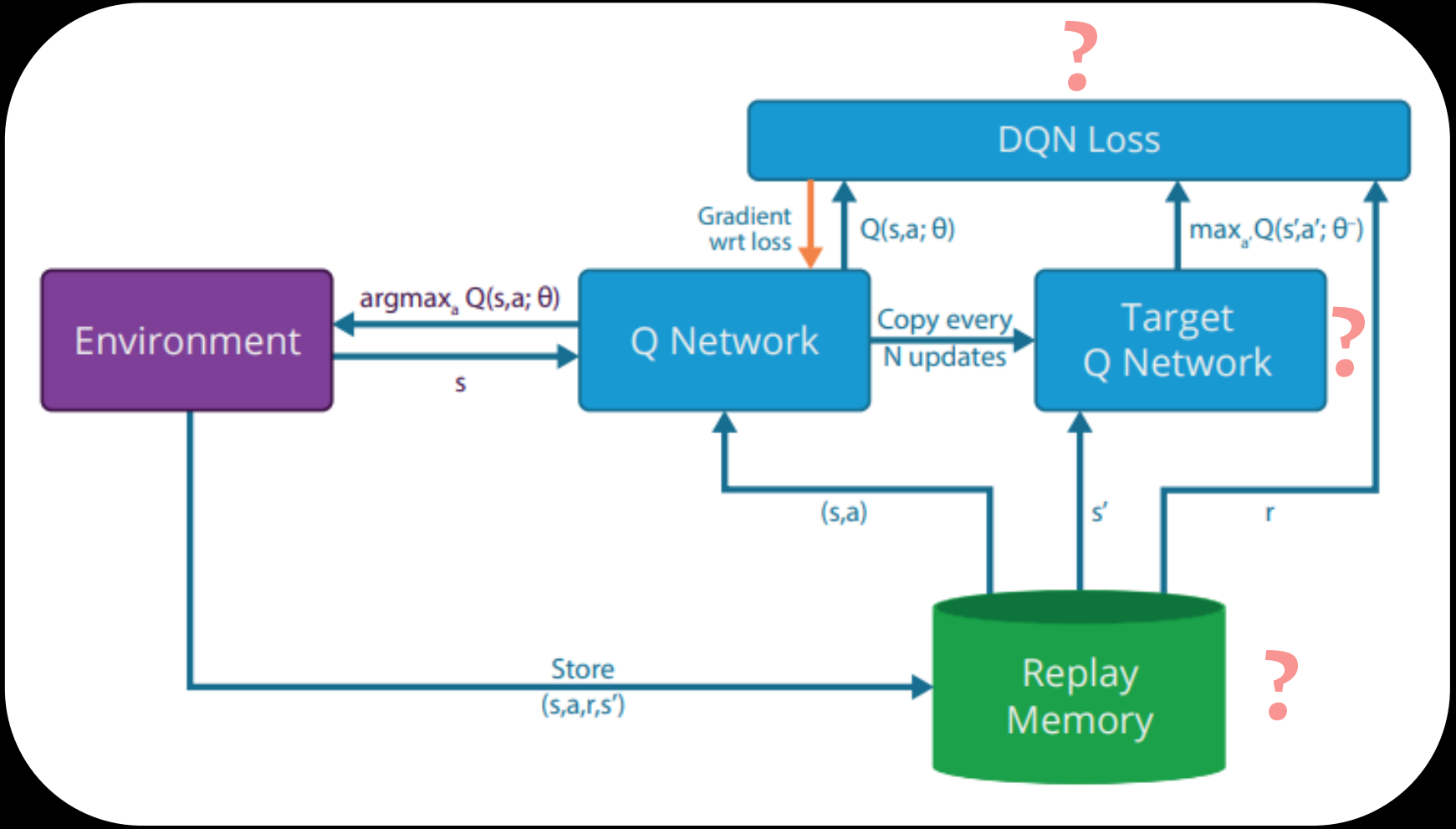
What are Q Values?

What if I move my knight to E5?

$$Q(s, a)$$

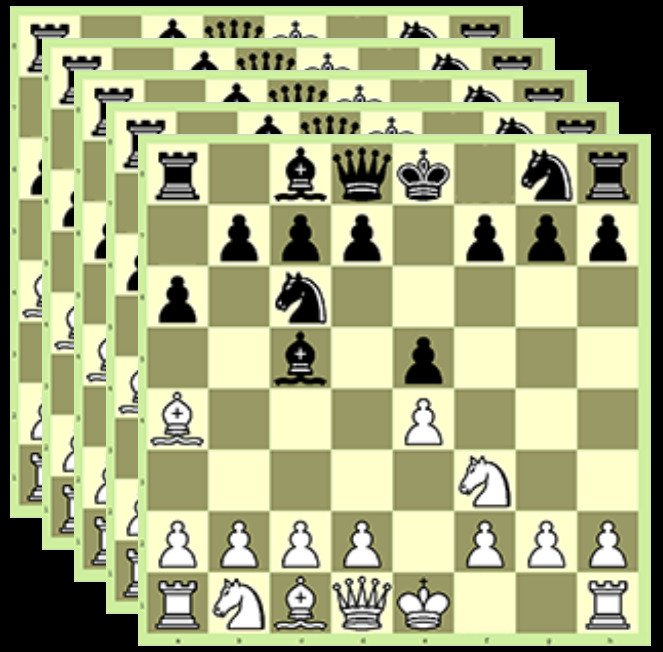
The expected **total** reward I will get…

We need to learn a few more terminologies



Let’s talk about replay memory

Mini-batches of observations



Let’s talk about target network

- Target network Q' has separate parameters θ^- than the original DQN.
- The target network is periodically updated to the current Q-network parameters.
- Stabilizes training process by reducing oscillations and divergence.

Let's talk about DQN Loss

$$L(\theta) = E_{(s,a,r,s')}\left[\left(r + \gamma * \max_{a'} Q(s', a';\ \theta^{-}) - Q(s, a; \theta)\right)^2\right]$$

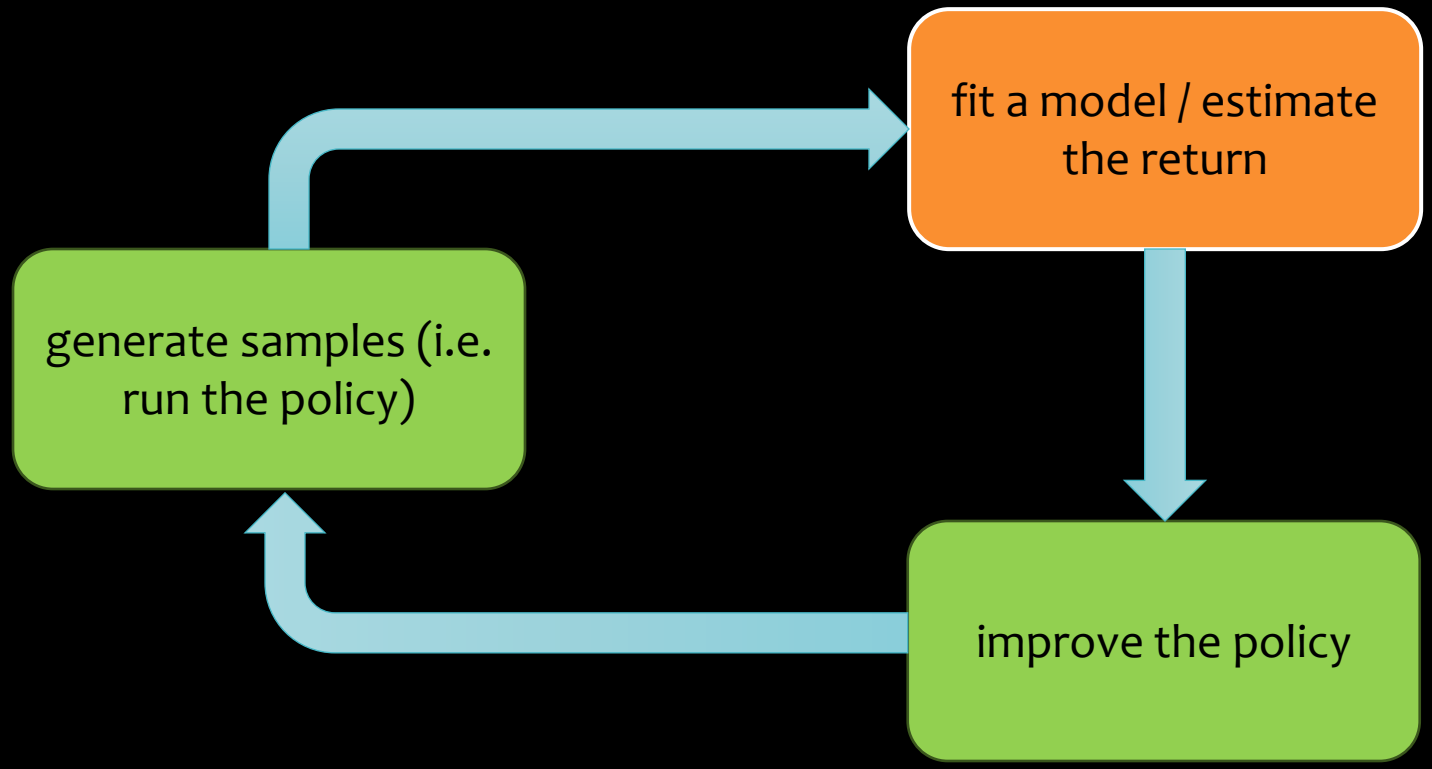
In practice expectations are just an indication that we need to sample data points

The estimate of the future reward based on target Q network

The estimate of the future reward by our DQN

Credits:

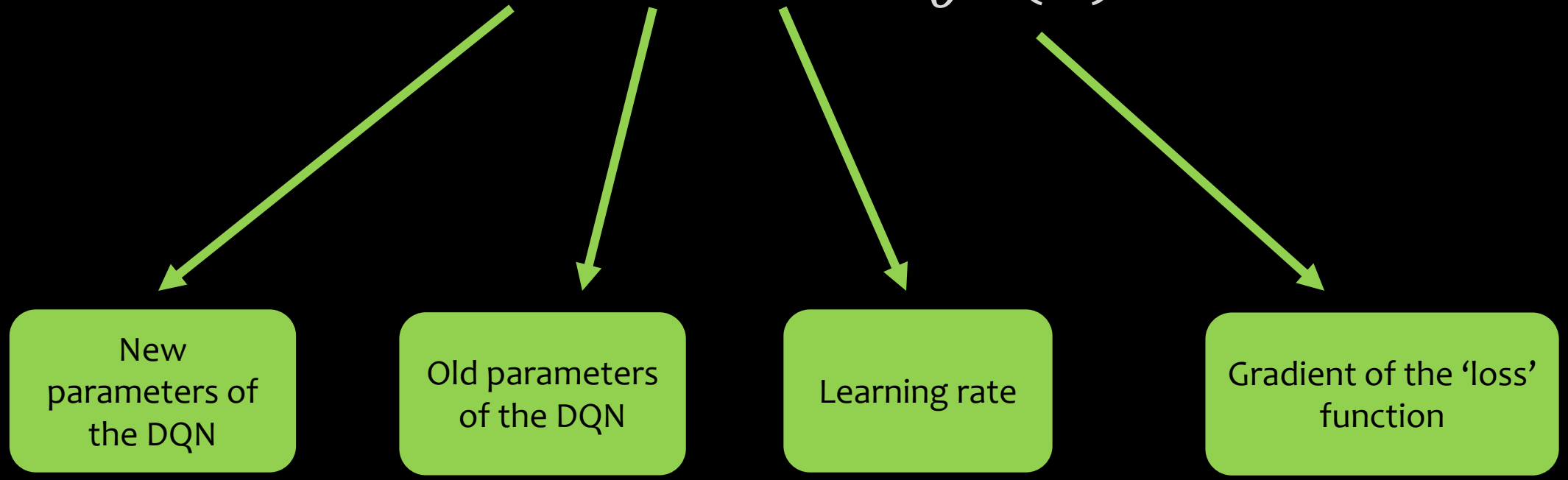
The trick to understand any RL algorithm



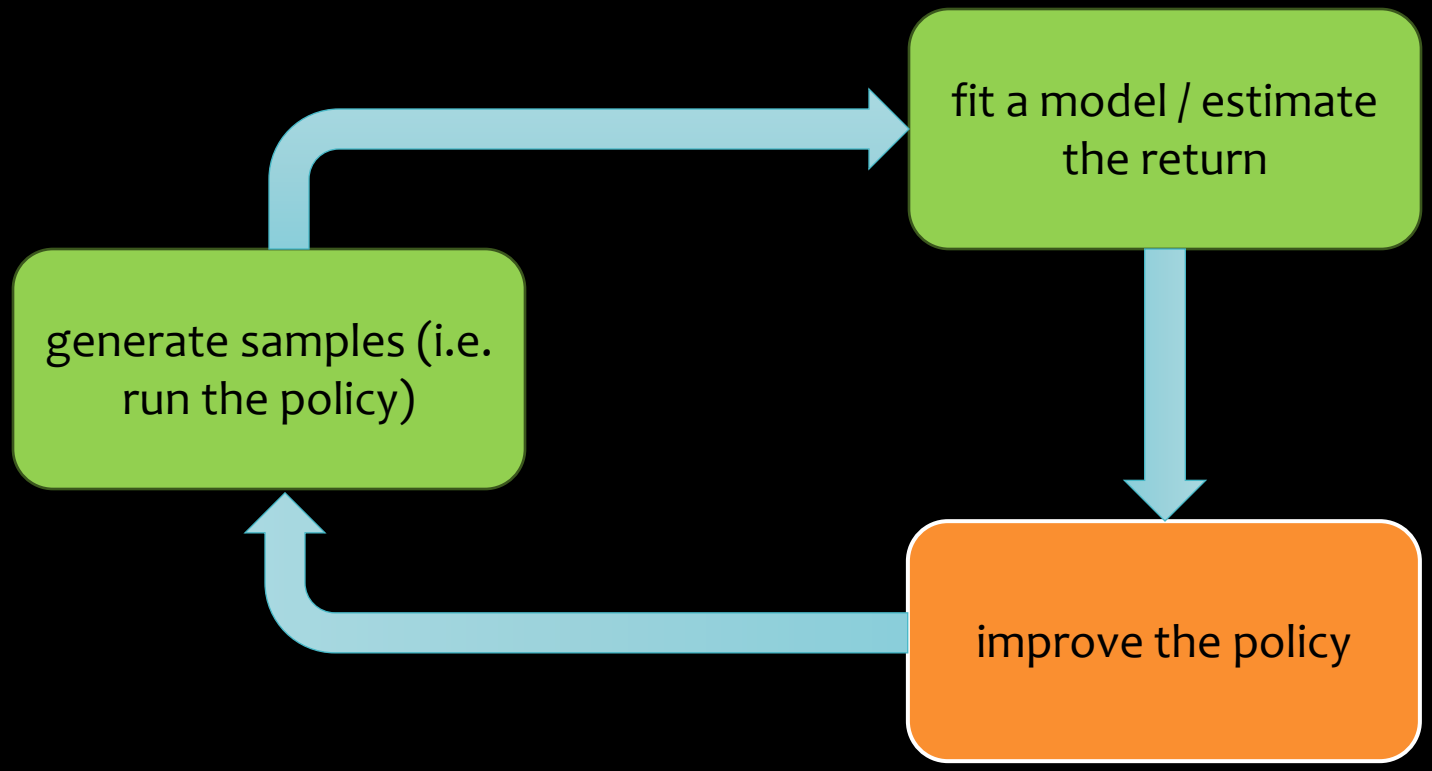
Source: UC Berkeley CS 285 – Lecture 4 Introduction to Reinforcement Learning

How to improve DQN Parameters?

$$\theta \leftarrow \theta - \alpha * \nabla_{\theta} L(\theta)$$

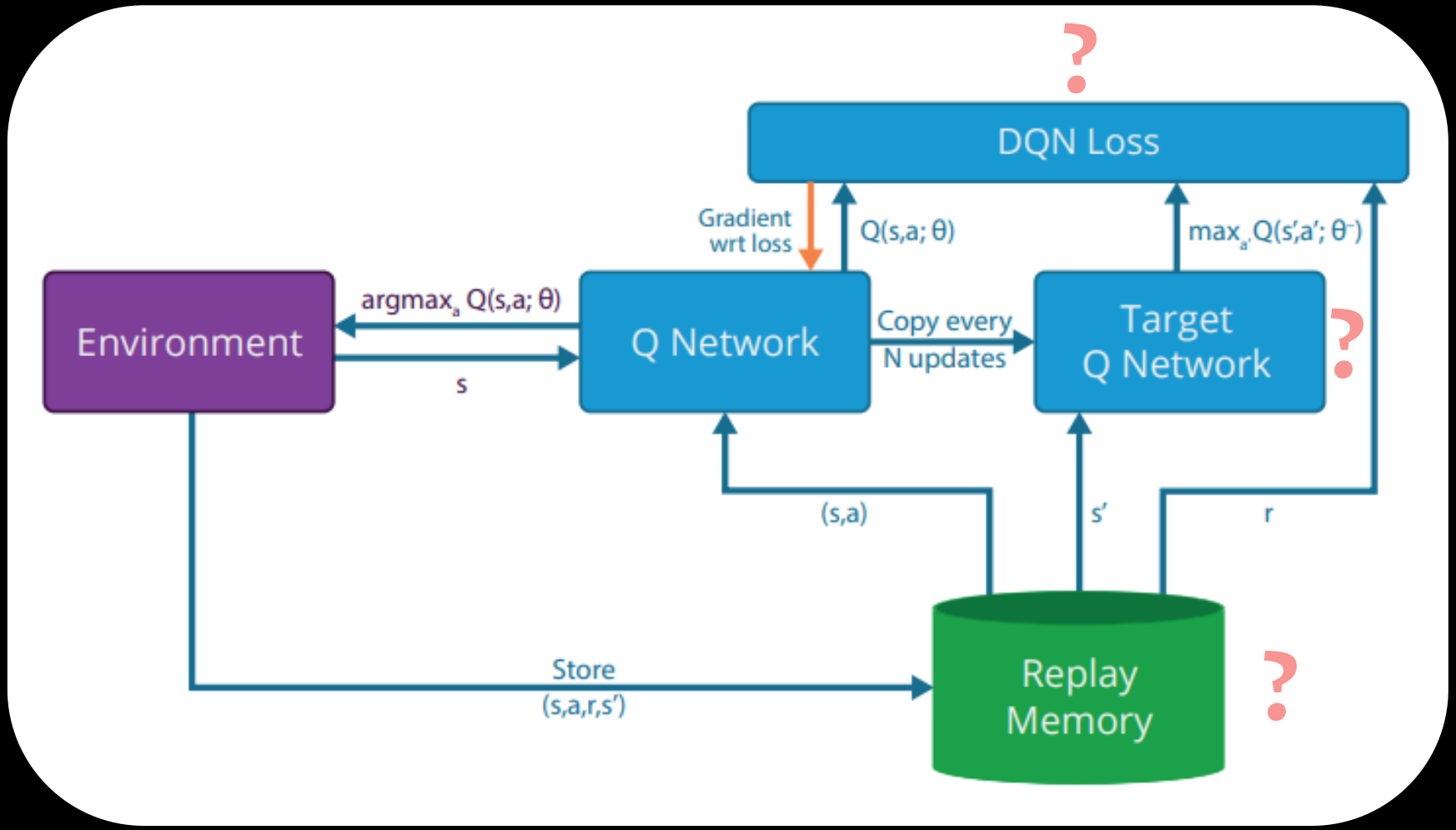


The trick to understand any RL algorithm



Source: UC Berkeley CS 285 – Lecture 4 Introduction to Reinforcement Learning

How it all connects!



Let’s have a break

Winner gets a KitKat

DQN Update

$$\theta \leftarrow \theta - \alpha * \nabla_\theta L(\theta)$$

Policy Gradient Update

$$\theta \leftarrow \theta + \alpha * \nabla_\theta J(\theta)$$

Why the difference?

Algorithm -> Business Value

- Design strategy for trading algorithms
- Optimize energy consumption in smart grids by learning from usage patterns
- Improve logistics and transportation planning by dynamically adjusting routes
- Optimize stock levels and reorder points by learning from historical sales data and demand patterns

Thank you for being here till the end!

As an engineer it is important to know the ‘how’

As a student of science, it is important to understand the ‘why’

Connect if you wish to...

Hetav Pandya

Software Engineer @ Arista Networks | University of Toronto Alumnus | Compute...

Feedback!

- Talk more about ML, seismology, physics, open-source software
- Teach me more about any topic
- Say hello, hola, namaste, ni hao
- Tag along for hiking, running, dancing ☺