AWS
BUILDERS' DAY

An Introduction to Reinforcement Learning

Julien Simon
Principal Technical Evangelist, AI & Machine Learning, AWS
@julsimon

Types of Machine Learning

Supervised learning

Run an algorithm on a labelled data set, i.e. a data set containing samples and answers. Gradually, the model learns how to correctly predict the right answer. Regression and classification are examples of supervised learning.

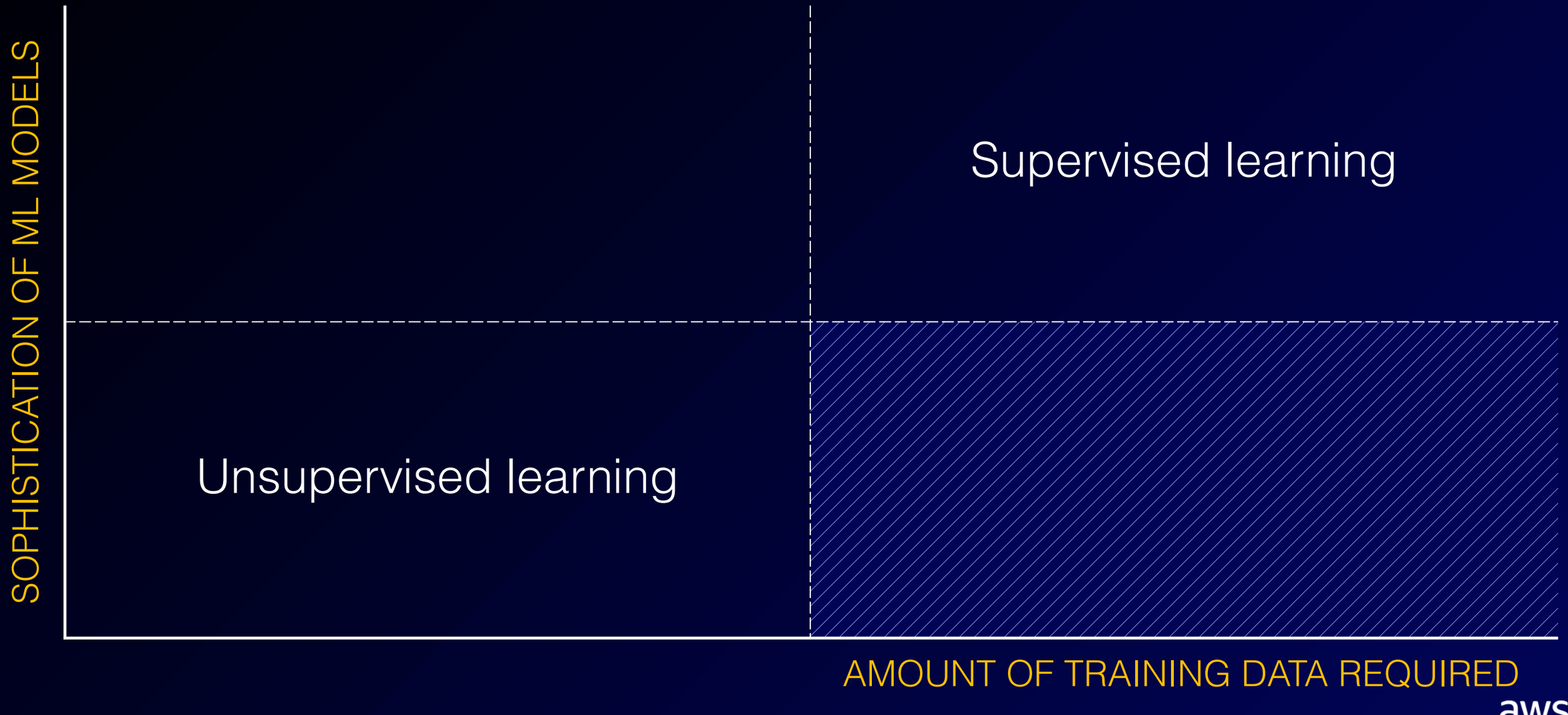
Unsupervised learning

Run an algorithm on an unlabelled data set, i.e. a data set containing samples only. Here, the model progressively learns patterns in data and organizes samples accordingly. Clustering and topic modeling are examples of unsupervised learning.

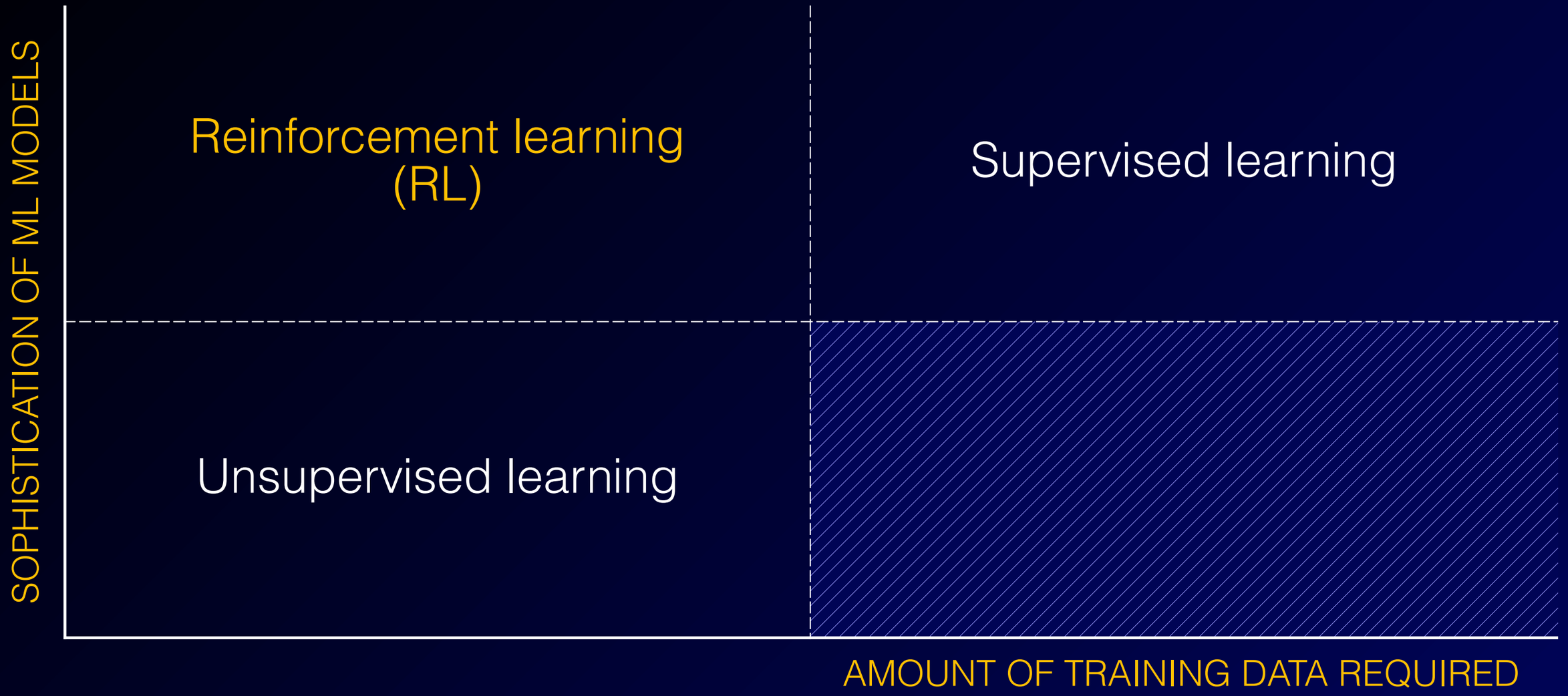
aws

Types of Machine Learning

Types of Machine Learning



Types of Machine Learning



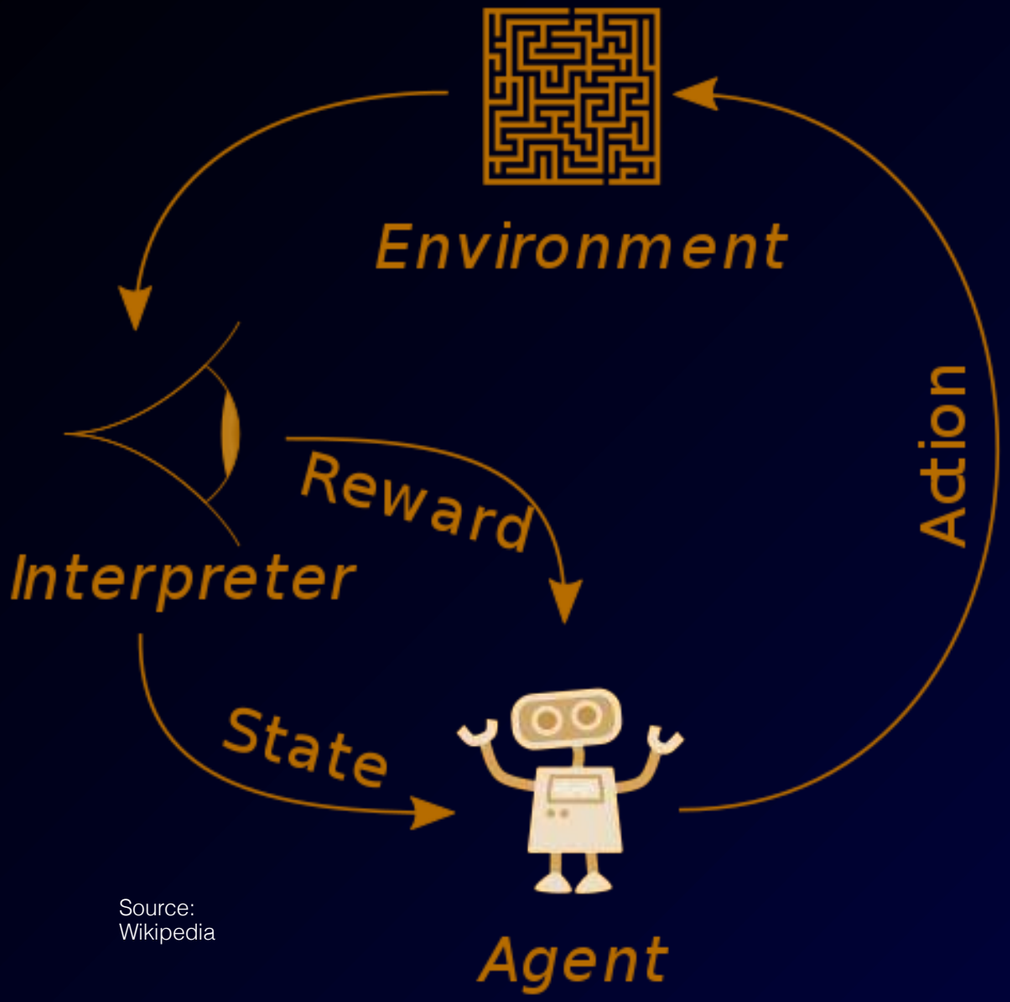
Remember when you first learned this?

We didn’t have an extensive labelled data set back then ☺

And yet we learned

How?

Defining Reinforcement Learning



Source:
Wikipedia

An algorithm (aka an agent) interacts with its environment.

The agent receives a positive or negative reward for actions that it takes: rewards are computed by a user-defined function which outputs a numeric representation of the actions that should be incentivized.

By trying to maximize the accumulation of rewards, the agent learns an optimal strategy (aka policy) for decision making.

aws

Use cases

- Large complex problems
  - Uncertain, dynamic environments
  - Continuous learning

- Supply chain management
- HVAC systems
- Industrial robotics
- Autonomous vehicles
- Portfolio management
- Oil exploration
- etc.

Caterpillar: 250-ton autonomous mining trucks
https://diginomica.com/2017/04/17/sending-disruption-mines/
https://www.cat.com/en_US/articles/customer-stories/built-for-it/thefutureisnow-driverless.html

Example: navigating a maze

- Imagine an agent learning to navigate a maze. It can move in certain directions but is blocked from going through walls.
- The agent discovers its environment (the current maze) one step at at time, receiving a reward each time: stepping into a dead end is a negative reward, moving one step closer to the exit is a positive reward.
- After a certain number of steps (or if we found the exit), the current episode ends.
- After a certain number of episodes, the agent uses the action/reward data points to train a model, in order to make better decisions next time around.
- One critical thing to understand is that the RL model isn't trained on a predefined set of labelled mazes (that would be supervised learning).
- This cycle of exploring and training is central to RL: given enough mazes and enough training time, we would soon enough know how to navigate any maze.

aws

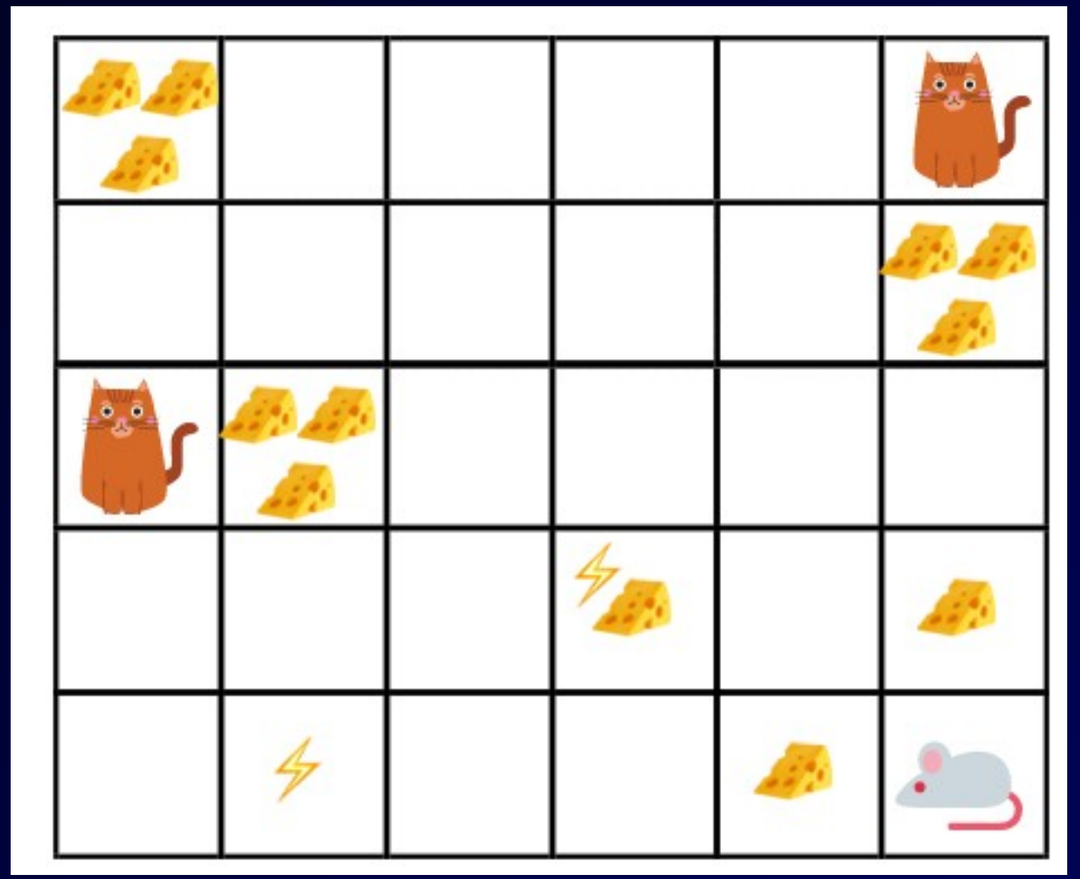
Environment

- The space in which the RL model operates.
- This can be either a real-world environment or a simulator.
- If you train a physical autonomous vehicle on a physical road, that would be a real-world environment.
- If you train a computer program that models an autonomous vehicle driving on a road, that would be a simulator… probably a much safer option!

aws

Exploitation vs Exploration

- Selecting the next action is a balance between exploitation (‘using what you’ve learned’) and exploration (‘taking a chance to learn new things’)
- If you favor exploitation, you may never reach high-value rewards.
- If you favor exploration, you’ll probably run into trouble very often!
- Initially, the agent will explore at random for a fixed number of episodes (aka heatup phase): this generates data for the first round of training.



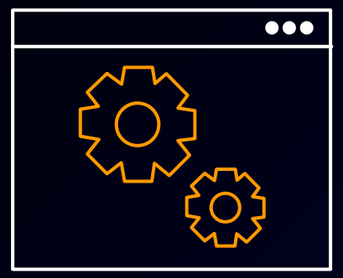
aws

Training a RL model

1. Formulate the problem: goal, environment, state, actions, reward
2. Define the environment: real-world or simulator?
3. Define the presets
4. Write the training code and the value function
5. Train the model

Amazon SageMaker RL

Reinforcement learning for every developer and data scientist



Fully managed

Broad support for frameworks

Broad support for simulation environments including SimuLink and MatLab

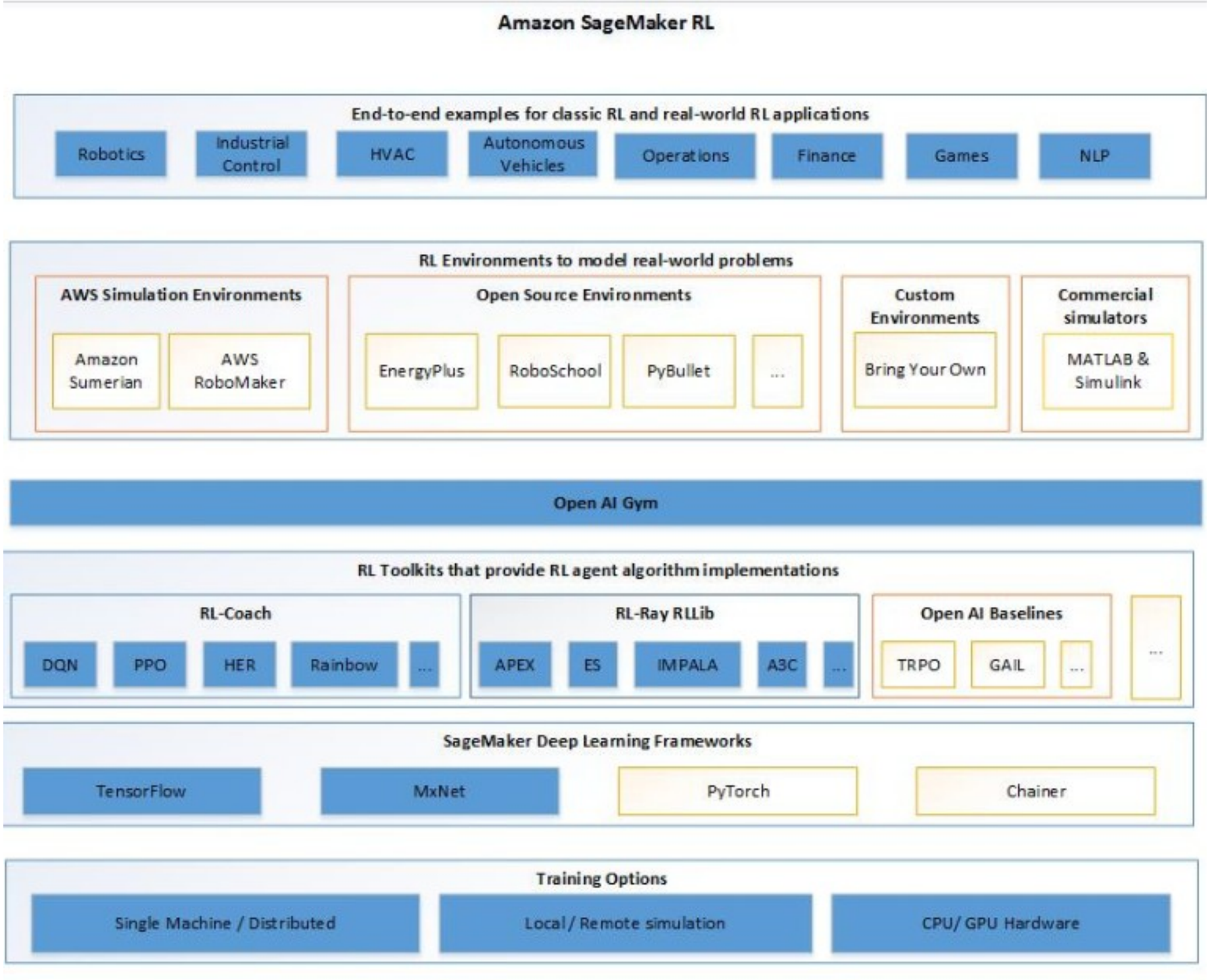
KEY FEATURES

TensorFlow, Apache MXNet, Intel Coach, and Ray RL support

2D & 3D physics environments and OpenAI Gym support

Supports Amazon Sumerian and Amazon RoboMaker

Example notebooks and tutorials


Amazon SageMaker RL
End-to-end examples for classic RL and real-world RL applications
Robotics
Industrial Control
HVAC
Autonomous Vehicles
Operations
Finance
Games
NLP
RL Environments to model real-world problems
AWS Simulation Environments
Amazon Sumerian
AWS RoboMaker
Open Source Environments
EnergyPlus
RoboSchool
PyBullet
...
Custom Environments
Bring Your Own
Commercial simulators
MATLAB & Simulink
Open AI Gym
RL Toolkits that provide RL agent algorithm implementations
RL-Coach
DQN
PPO
HER
Rainbow
...
RL-Ray RLLib
APEX
ES
IMPALA
A3C
...
Open AI Baselines
TRPO
GAIL
...
...
SageMaker Deep Learning Frameworks
TensorFlow
MxNet
PyTorch
Chainer
Training Options
Single Machine / Distributed
Local / Remote simulation
CPU/ GPU Hardware

Demo – Cartpole

Apache MXNet + Intel Coach + OpenAI Gym

Demo – Knapsack

Apache MXNet + Intel Coach + custom environment

Demo – Predictive Auto Scaling

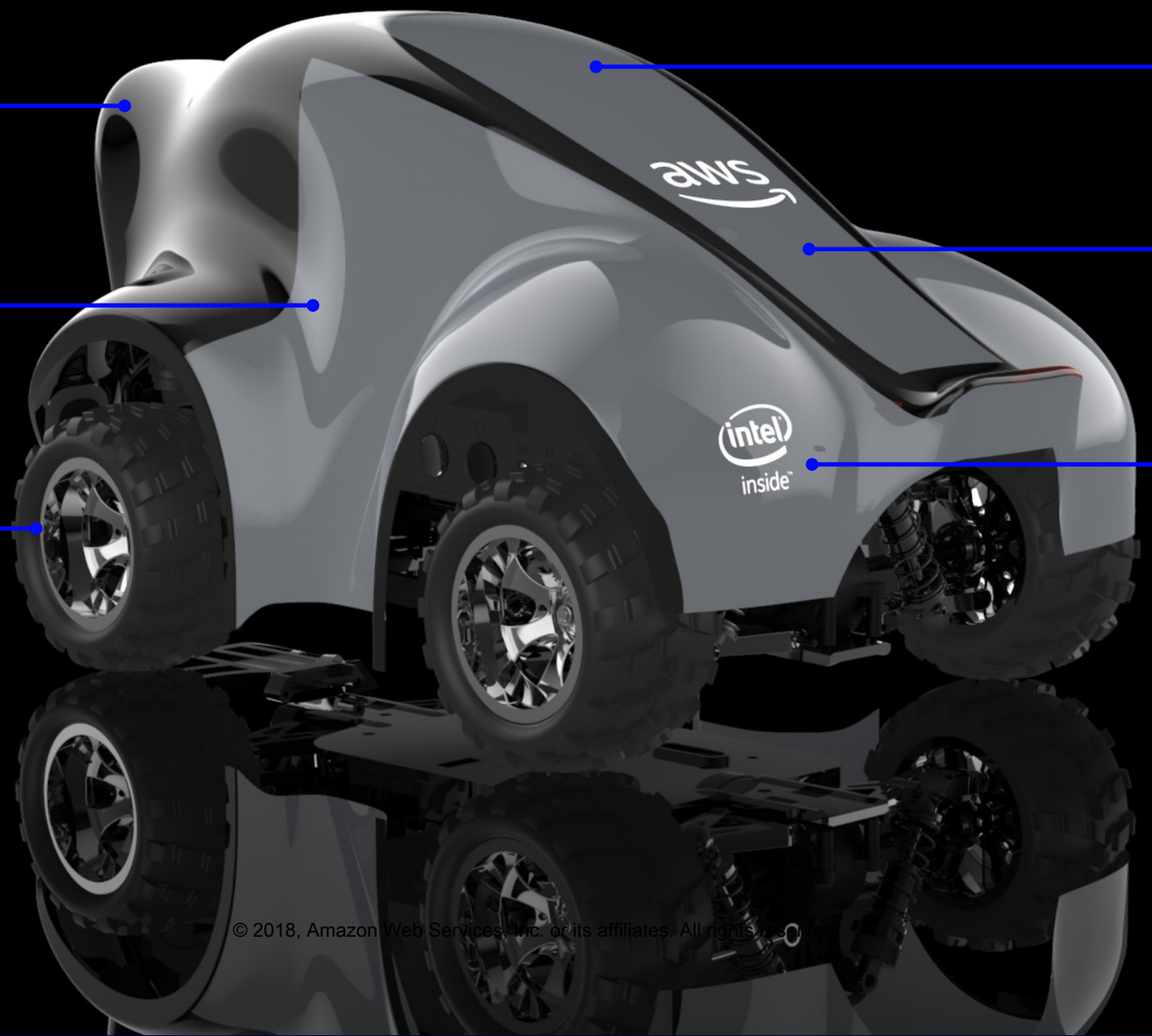
TensorFlow + Intel Coach + custom environment

How can we get developers rolling with reinforcement learning?

Introducing AWS DeepRacer

Fully autonomous 1/18th scale race car, driven by reinforcement learning

https://youtu.be/X-6v4RZy-TE



HD video camera

Gyroscope

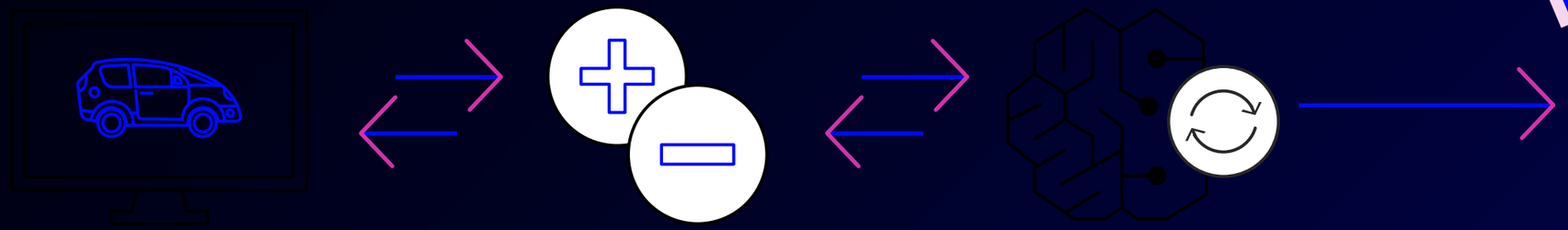
Four-wheel drive

Dual power for compute and drive

Accelerometer

Dual-core Intel processor

AWS DeepRacer



Simulation environment

Scoring function

RL algorithm

AWS DeepRacer League

Competitive racing league for AWS DeepRacer

Train models with RL

Compete virtually online

Race in trials

Final at AWS re:Invent

Getting started

http://aws.amazon.com/free

https://ml.aws
https://aws.amazon.com/sagemaker
https://aws.amazon.com/deepracer/

https://github.com/aws/sagemaker-python-sdk
https://github.com/awslabs/amazon-sagemaker-examples

https://medium.com/@julsimon
https://gitlab.com/juliensimon/dlnotebooks

Thank you!

Julien Simon
Principal Technical Evangelist, AI & Machine Learning, AWS
@julsimon