Machine Learning, Reinforcement Learning

Reinforcing the science behind Reinforcement Learning

Dummies guide to Reinforcement learning, Q learning, Bellman Equation.

You're getting bore struck in lockdown, you decided to play computer games to pass your time.

You launched Chess and chose to play against the computer, and you lost!

But how did that happen? How can you lose against a machine that came into existence like 50 years ago?

Photo by Piotr Makowski on Unsplash

This is the magic of **Reinforcement learning.**

Reinforcement learning lies under the umbrella of **Machine Learning.** They aim at developing intelligent behavior in a complex dynamic environment. Nowadays since the range of AI is expanding enormously, we can easily locate their importance around us. These play an important role ranging from *Autonomous Driving, Recommender Search Engines, Computer games to Robot movements*.

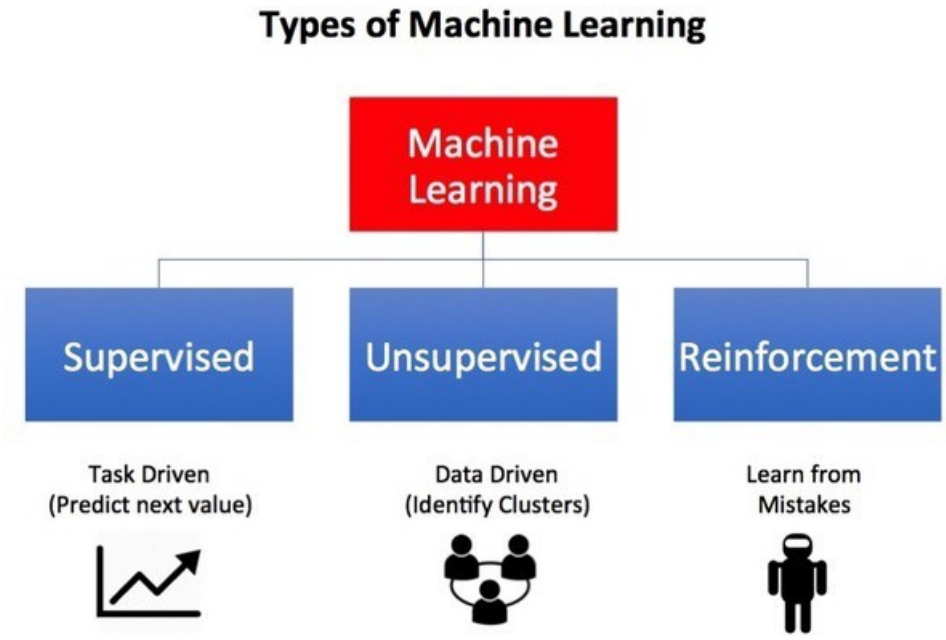
Pavlov's Conditioning

When we think about AI, we have a perception of thinking about the future, but our idea takes us back to more than 100 years ago. In the late 19th century, *Ivan Pavlov*, a Russian physiologist was studying the salivation effect in dogs. He was interested to know how much dogs were salivating when they would see the food, but, while conducting the experiment, he noticed that dogs were even salivating before seeing any food. After his conclusions on that experiment, Pavlov would ring a bell before feeding them and as expected they again started salivating. The reason behind their behavior can be their ability to learn **because they had learned that after the bell, they'll be fed**. Another thing to ponder is, the dog doesn't salivate because the bell is ringing but because given past experiences he had learned that food will follow the bell.

Photo by engin akyurt on Unsplash

What is Reinforcement Learning?

Reinforcement Learning is a part of Machine Learning techniques that enables an AI agent to interact with the environment and thus learn from its own sequence of actions and experiences.



Source

For sake of illustration, imagine you're stuck at an isolated island. You can expect yourself to freak out first, but, with no options left, you'll start fighting for your survival. You'll look for a place to hunt, you'll look for a place to sleep, you'll inspect what to eat and what to avoid. If you stay at a safe place, you'll notice that it is correct action that has to be performed by you and, at the same instant, if you ate some animal that

led to diarrhea you'll avoid eating that in future. Your actions will become better over time and you'll easily adjust to the new environment by learning. Reinforcement learning follows the same method wherein we expect the agent to experience the new environment, track its actions and consequences by discovering errors and rewards, and learn to get better or aims at maximizing the reward.

Photo by Aleks Dahlberg on Unsplash

But, how does it compare against supervised learning?

It is possible to use a supervised learning method instead of reinforcement learning techniques. But, for that, we need a really large dataset that would constitute every action and its consequence. Its next unfavorable outcome would be limited learning, suppose if track actions of best player but still he is not perfect and following his actions machine might become great like him but won't be able to exceed his scores.

	Supervised Learning	Reinforcement Learning
Definition	Works on existing data	Aims at interacting with environment
Task	Classification and Regression	Exploitation and Exploration
Mapping between Input and Output	Both input and output will contribute in decision making and model will be trained on given sample data	Will get constant feedback from the input as form of reward/penalty and thus making its actions better.

And, how does it stand against Unsupervised Learning?

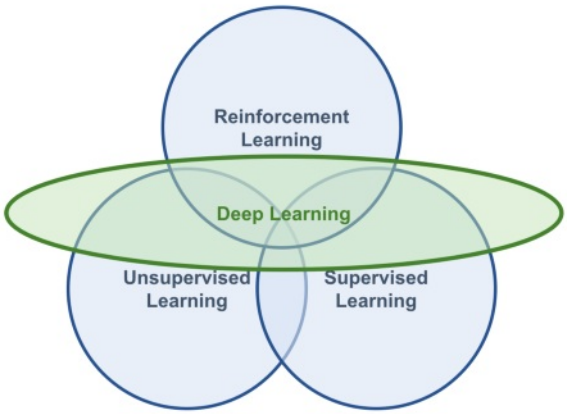
In unsupervised learning, there is no direct connection between input and output rather it aims at recognizing the patterns, on the contrary, Reinforcement learning is all about learning from the output provided by past input.

	Supervised Learning	Reinforcement Learning
Definition	No pre-trained data provided	Aims at interacting with environment
Task	Clustering	Exploitation and Exploration
Mapping between Input and Output	It aims at looking for pattern rather than the mapping	Will get constant feedback from the input as form of reward/penalty and thus making its actions better.

Then, is it Deep Learning?

Deep learning irrefutably comes under the umbrella of Machine Learning and is capable of computing complex problems that require human-like intelligence.

The Venn-diagram shows the relation between all Machine Learning techniques, according to Universal Approximation Theorem(UAT), we can solve any problem using Neural Nets, but these are not necessarily an optimal solution to every problem as they require a lot of data to process and are often challenging to interpret.



Analyzing the figure, it shows that we are not required to use Deep Learning for every Reinforcement Learning problem that clears the myth that **it doesn't solely depend upon Deep Learning**.

How does Reinforcement Learning work?

In Reinforcement Learning, we aim at the interaction of Agent and Environment.

- An **Agent** can be regarded as the “solution”, which is a computer program that we expect to make decisions to solve decision-making problems.
- An **Environment** can be regarded as the “problem”, which is where the decision taken by the agent is implemented.

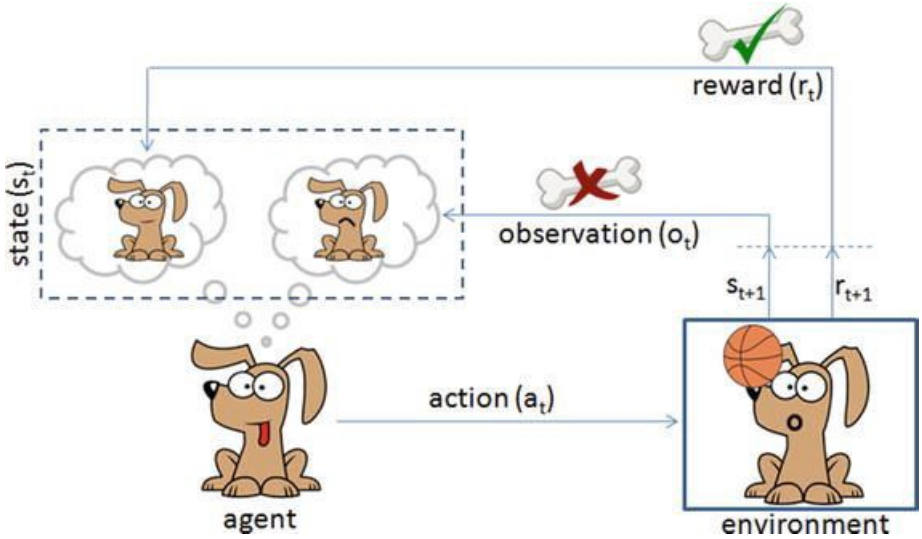
For example, in the case of the chess game, we can consider that the Agent is one of the players and the Environment constitute the board and competitor.

Both components are inter-dependent in a way that the Agent tries to adjust its actions based on the influence by the Environment, and Environment reacts to Agent’s action.

The Environment is bound by a set of variables that are usually associated with decision-making problems. A set of all possible values can be regarded as **state space**. A **state** is a part of state space i.e. a value the variable takes.

At each state, the Environment is entitled to provide a set of actions to the Agent, amongst whom it should choose one. The agent tries to influence the Environment using these actions and Environment may change states as a response to the Agent’s actions. **Transition function** is something that tracks these associations.

The Environment either reward or penalize the agent based on its actions. The **Reward** is the positive feedback provided if the last action of the agent is contributing to achieving a favorable goal. The **Penalty** is the negative feedback provided by the environment if the last action of the agent results in a deviation from the goal. The agent’s goal is to maximize the overall reward and keep making its actions better in order to achieve the desired final result.



Another thing that Reinforcement learning requires is a lot of training time, as the rewards aren't disclosed to the Agent until the end of an episode(game). e.g. if our computer is playing chess against us and it wins, then it will be rewarded (as our desired outcome was to *win*) but still, it needs to figure out for which actions it was rewarded and that can only be achieved when it is given a tonne of training time and data.

How does Reinforcement Learning learn? (Q-learning)

Goal: To maximize the total reward

$$Q(s,a) \ \ where$$
$$Q: \ quality \ of \ action$$
$$s: \ State$$
$$a: \ Action$$

We expect, the rewards to come early as to make our training faster and thus quickly achieving desired outcomes.

$$R_t = r_t + r_{t+1} + r_{t+2} + ... + r_{t+n}$$
$$where \ R_t \ is \ Total \ Reward.$$

Ideal Case

But, in real case we encounter late rewards and to penalize late rewards we will introduce Discount Factor().

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + ... + \gamma^{n-1} r_{t+n}$$

where R_t is Total Reward.

γ is Discount Factor.

Real Case

In real case, as we move towards right, the uncertainty increases.

Algorithm Q-learning

Input: policy π, positive integer *num_episodes*, small positive fraction α, GLIE $\{\epsilon_i\}$
Output: value function Q ($\approx q_\pi$ if *num_episodes* is large enough)
Initialize Q arbitrarily (e.g., $Q(s,a) = 0$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$, and $Q(terminal\text{-}state, \cdot) = 0$)
for $i \leftarrow 1$ ***to*** *num_episodes* **do**
  $\epsilon \leftarrow \epsilon_i$
  Observe S_0
  $t \leftarrow 0$
  repeat
    Choose action A_t using policy derived from Q (e.g., ϵ-greedy)
    Take action A_t and observe R_{t+1}, S_{t+1}
    $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t))$
    $t \leftarrow t + 1$
  until S_t *is terminal*;
end
return Q

Q-learning

Bellman Equation

$$Q(s,a) = r + \gamma * max(s', a')$$

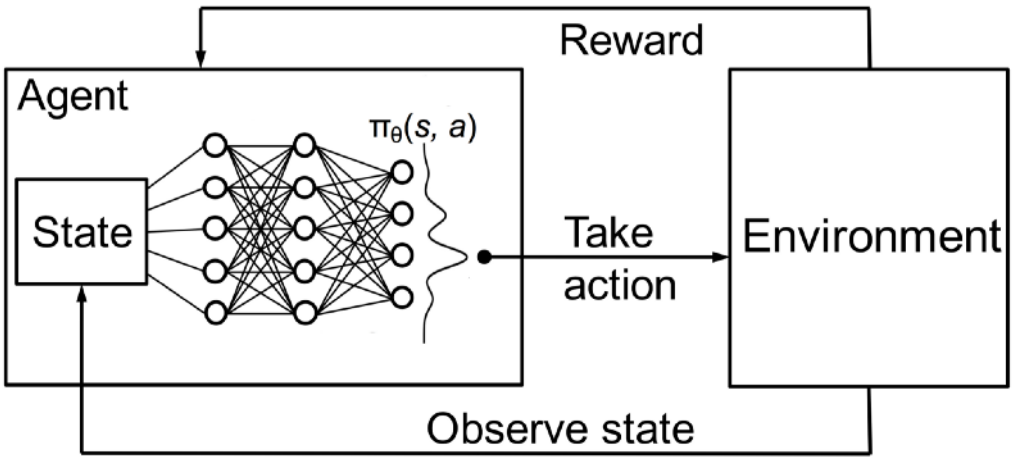
where Q = *Quality of action*

r = *Reward*

γ = *Discounting Factor*

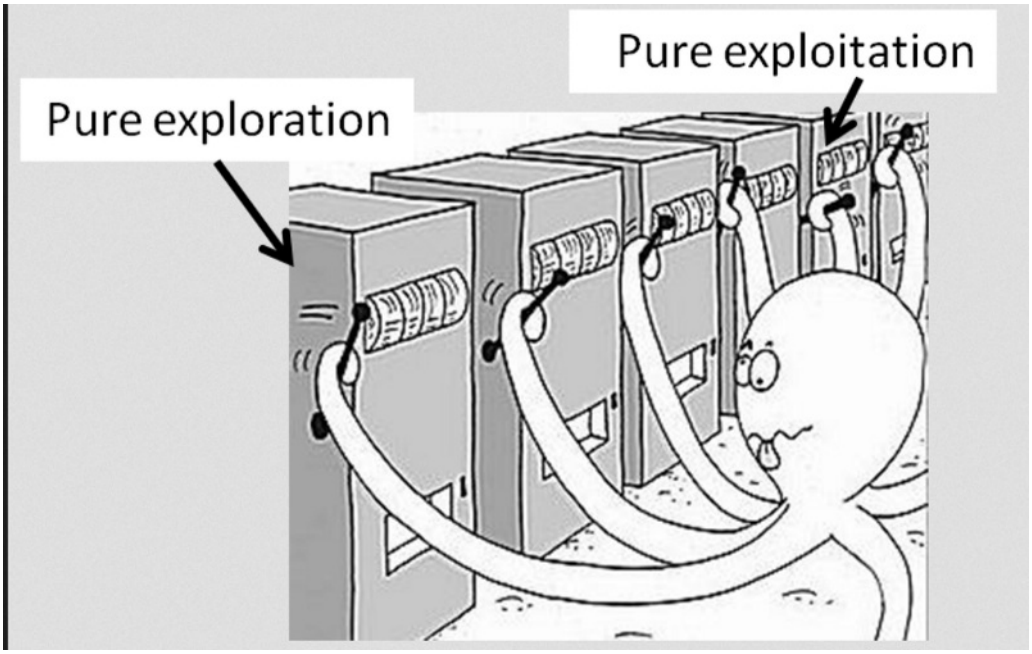
Our goal was to maximize the reward or we can say to minimize the error(loss).

To minimize the loss, we can implement Gradient Descent using Mean-square error loss.



Exploration v/s Exploitation trade-off

Other interesting components of Reinforcement Learning are **Exploration** and **Exploitation**. To obtain quick rewards, an Agent must follow past experiences. But to detect such actions, it has to try actions at first.



Exploration vs Exploitation, Source

In nutshell, to obtain quick rewards an Agent has to *exploit* but it is also expected to explore to make its actions better, and thus that might help it to get a better reward.

Let's get back to the island, you've three spots for fishing and each is home to three types of fishes, spot 1 is habitat to *black fishes which are* poisonous, spot 2 is home to *orange fishes* that are delicious as well as nutritious and, spot 3 constitutes *grey fishes* that are best in terms of nutrition and taste. The goal would be not to eat blackfish and try to have a grey one.

Spot1 vs Spot2 vs Spot3

Let's assume, on Day 1, you chose spot 1 for fishing and ended up eating a blackfish and diarrhea. On Day 2, you reached spot 2 and ended up having a delicious meal. Now, your instincts will try to exploit the path

you've chosen i.e. the road to spot 2 because for your mind going to spot 2 is better policy based on the first two episodes. And, hence, your mind will be stuck in a policy where it is sacrificing for a moderate award.

> **Exploration**: Helps you to try various actions; good in the beginning.
>
> **Exploitation**: Sample good experience from past; need memory space; good at the end

Conclusion

Hopefully, this article will help you to understand about Reinforcement Learning in the best possible way and also assist you to its practical usage.

As always, thank you so much for reading, and please share this article if you found it useful!

Feel free to connect:

> LinkedIn ~ https://www.linkedin.com/in/dakshtrehan/
>
> Instagram ~ https://www.instagram.com/_daksh_trehan_/
>
> Github ~ https://github.com/dakshtrehan

Follow for further Machine Learning/ Deep Learning blogs.

> Medium ~ https://medium.com/@dakshtrehan

Want to learn more?

Detecting COVID-19 Using Deep Learning

The Inescapable AI Algorithm: TikTok

An insider's guide to Cartoonization using Machine Learning

Why are YOU responsible for George Floyd's Murder and Delhi Communal Riots?

Decoding science behind Generative Adversarial Networks

Understanding LSTM's and GRU's

Recurrent Neural Network for Dummies

Convolution Neural Network for Dummies

Diving Deep into Deep Learning

Why Choose Random Forest and Not Decision Trees

Clustering: What it is? When to use it?

Start off your ML Journey with k-Nearest Neighbors

Naive Bayes Explained

Activation Functions Explained

Parameter Optimization Explained

Gradient Descent Explained

Logistic Regression Explained

Linear Regression Explained

Determining Perfect Fit for your ML Model

Cheers