Intro To AI (CSCI - 6600)
Fall 2022

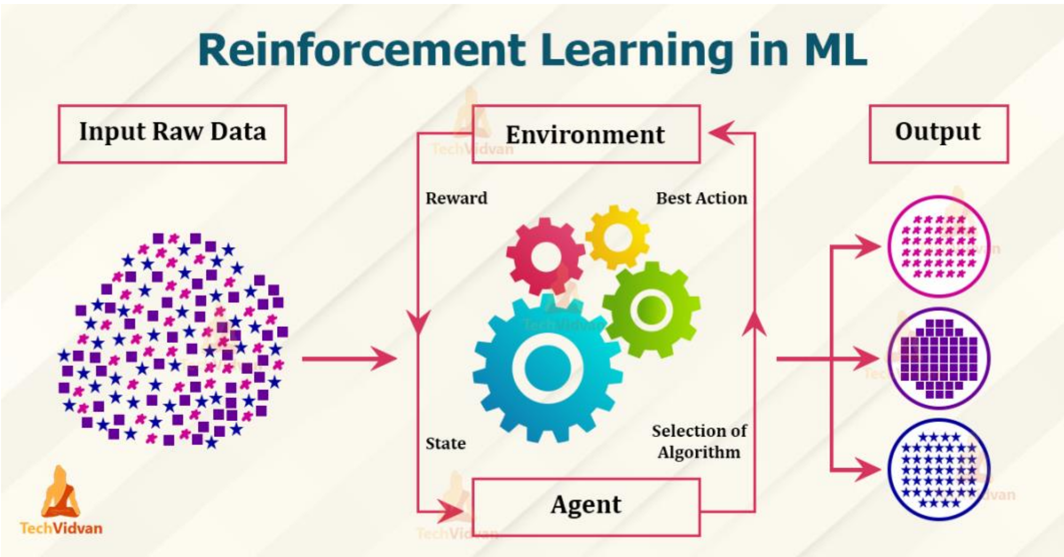
A Comparison Between Model-Less And Model-Based RL Approaches

By

Rajdeep Bhattacharya **Adesh Agarwal**

Pranay Udaygiri **Rajwinder Singh**

Comparison And Evaluation



“We couldn’t do parallel simulation due to limitations in our laptop capabilities.”

Having played this game in both Model-Less and Model-Based approaches, here are some differences that we have noticed: -

1. In Model-Free the **rewards** are not accounted for (since this is automated, reward = 1) but in Model-Based rewards are accounted for.

2. In Model-Free no modelling (no **decision policy** is required) but in Model-Based modelling is required (**policy network**).

3. Model-Free doesn’t require the use of **initial states** to predict the next state but Model-Based requires the use of initial states to predict the next state using the policy network.

4. In Model-Free the **rate of missing the ball** with respect to time is zero but in Model-Based the rate of missing the ball with respect to time approaches zero.

5. **Variance** - A discrete policy produces more or less same in different runs so less variance over different runs.

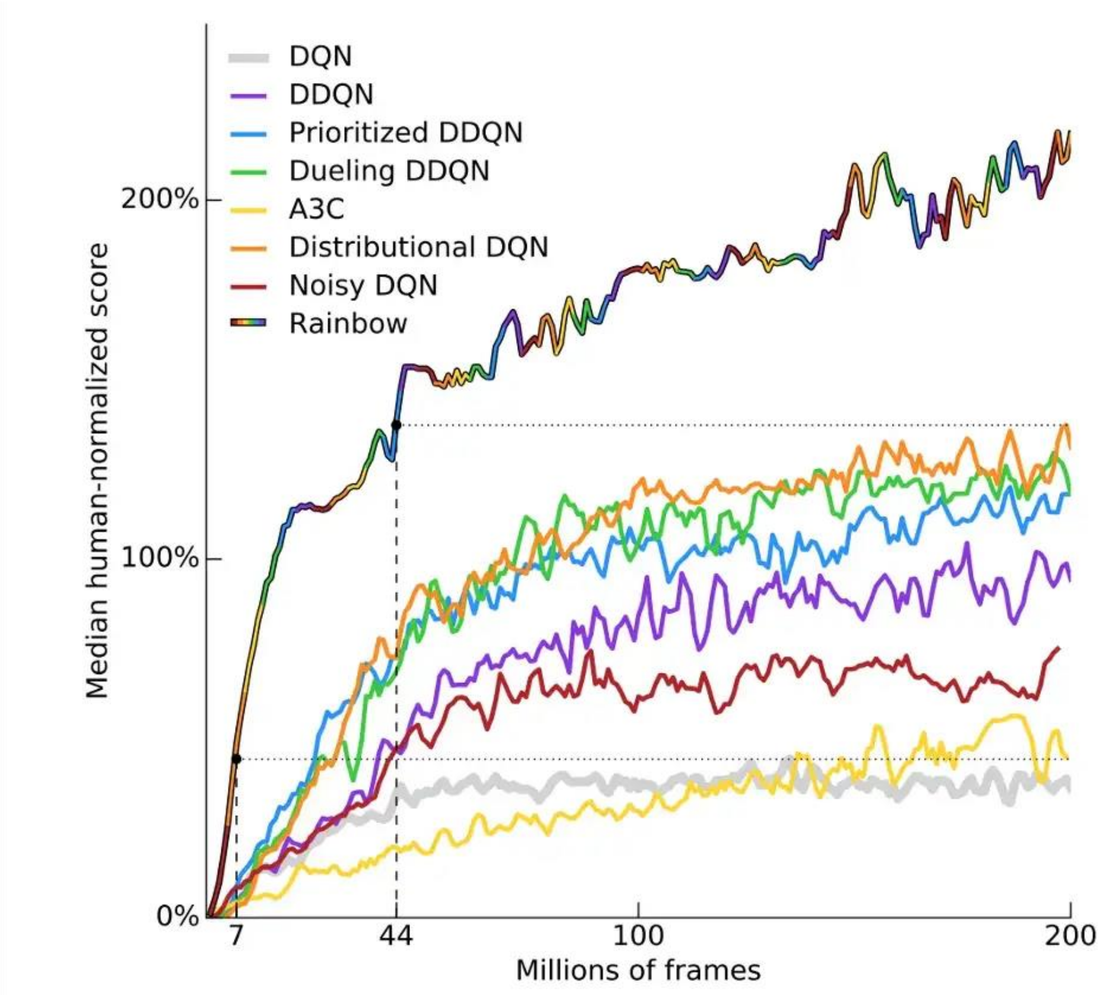
6. Assumption 1 - In value fitting methods and some model-based methods, we assume the action space and/or the **state space is continuous.**

7. Assumption 2 - QN is mainly for **low-dimensional discrete control space**.

8. DQN is an off-policy method, **samples are drawn from a replay buffer to fit the Q-value**. In DQN, the replay buffer improves both stability and sample efficiency.

Here is the plot on the performance of many **Model-Less Methods** from the Internet.

As noted, it easily takes 80+ million frames for many advanced methods to outperform the human expert in playing Atari games. The graph below is normalized. An average human expert score at 100% below.



Conclusion

There were a few problems that we ran into, like as the Tensor Flow package is heavy in size it took time to install and again after that it was difficult to run on Macbooks. Also there was a minor issue with the PyGame version we were initially using.

Tennis might be simple compared to self-driving cars, but hopefully this example showed us a few things about RL.

The main difference between Model-Less and Model-Based RL is the policy network, which is required for Model-Based RL and unnecessary in Model-Less.

It's worth noting that oftentimes, Model-Based RL takes a massive amount of time for the Deep Neural Networks (DNNs) to learn the states perfectly without getting it wrong.

It is more like Tenured Job and Gig hehe, the first one needs more preparation but is more reliable.

But every technique has its drawbacks and advantages, choosing the right one depends on what exactly you need your program to do.

Model-Based and Model-Less RL Methods – Strengths and Weaknesses

	Strength	Weakness
Model-Based	It can be self-trained and therefore more scaleable. Sample efficient. The learned dynamic model is transferable.	But need retraining to optimize the controller again for a specific task. Not optimizing the policy directly. More assumptions and does not work with all tasks. A model can be much complex to train than a policy.
Model-Less	It has fewer approximations and assumptions that work with a wider spectrum of tasks. Good at learning complex policies. The policy can be generalized better in some tasks.	Less sample efficient. Vulnerable to overfit with a complex model. Lead to poor decisions.

Model-Based Vs Model-Less RL Methods – Strength And Weakness

We could have done a more robust study had we had more time. Among the different ML options some of us have already become fan of RL and we hope to explore more about it in the future.

Thank you.