

Joint Power Allocation and Network Slicing in an Open RAN System

Mojdeh Karbalaee Motalleb

School of ECE, College of Engineering, University of Tehran, Iran

Email: {mojdeh.karbalaee}@ut.ac.ir,

Abstract—
Index Terms—

I. QUESTIONS

1) Why deep Q learning is better than Q learning?

Deep RL uses a Deep Neural Network to approximate $Q(s,a)$. Non-Deep RL defines $Q(s,a)$ using a tabular function.

Popular Reinforcement Learning algorithms use functions $Q(s,a)$ or $V(s)$ to estimate the Return (sum of discounted rewards). The function can be defined by a tabular mapping of discrete inputs and outputs. However, this is limiting for continuous states or an infinite/large number of states. A more generalized approach is necessary for large number of states.

Function approximation is used for a large state space. A popular function approximation method is Neural Networks. You can make a Deep Neural Network by adding many hidden layers.

Thus, Deep Reinforcement Learning uses Function Approximation, as opposed to tabular functions. Specifically DRL uses Deep Neural Networks to approximate Q or V (or even A).

When the environment gets complicated, the knowledge space can become huge and it no longer becomes feasible to store all (state, action) pairs. If you think about it in raw terms, even a slightly different state is still a distinct state (e.g. different position of the enemy coming through the same corridor). You could use something that can generalize the knowledge instead of storing and looking up every little distinct state.

So, what you can do is create a neural network, that e.g. predicts the reward for an input (state, action) (or pick the best action given a state, however you like to look at it) So, what you effectively have is a NN that predicts the Q value, based on the input (state, action). This is way more tractable than storing every possible value like we did in the table above.

$Q = \text{neural-network.predict}(\text{state}, \text{action})$

2) Find QoS that is required for different applications

- **Bandwidth and throughput:** Bandwidth is the available capacity of connection between two terminals as the most popular term for that is (bps). Throughput slightly differs from bandwidth as it stands for effective bandwidth that is provided by network

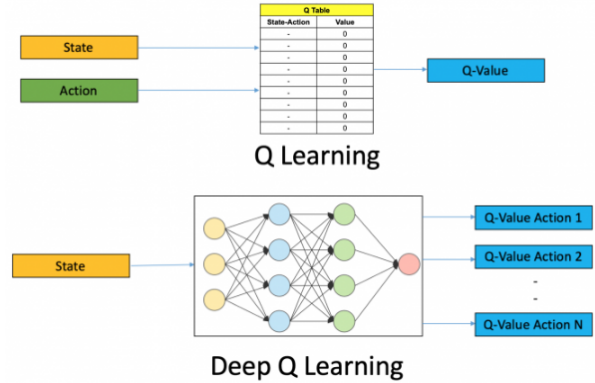


Fig. 1: deep Q learning structure [1]

Application	Bandwidth	Sensitivity to:		
		Delay	Jitter	Loss
VOIP	Low	High	High	Med
Video Conferencing	High	High	High	Med
Streaming Video	High	Med	Med	Med
Streaming Audio	Low	Med	Med	Med
Client/Server Transactions	Med	Med	Low	High
Email	Low	Low	Low	High
File Transfer	Med	Low	Low	High

Fig. 2: deep Q learning structure [2]

- **Delay or latency:** It specifies the time it takes for a packet to leave source until reaching the destination. Applications and network devices can cause delay.
- **Jitter (delay variation):** Jitter is an interval between subsequent packets. It is occurred by network congestion, route alternation and etc.
- **Loss:** It is amount of packets out of all that are not received at destination. The success of QoS depends on this factor.
- **Reliability:** Some applications are sensitive to packet loss such as real-time applications. Thus there must be some mechanism either in application or network to minimize the packet loss, such as forward error correction (FEC).

REFERENCES

- [1] (2020) Deep reinforcement learning: Guide to deep q-learning. [Online]. Available: <https://www.mlq.ai/deep-reinforcement-learning-q-learning/>
- [2] A. Gholamhosseinian, A. Khalifeh, and N. Zare Hajibagher, "Qos for multimedia applications with emphasize on video conferencing," 2011.