Spectrally-Efficient Multiple Access Design for Beyond 5G/6G Networks Exploiting Deep Reinforcement Learning

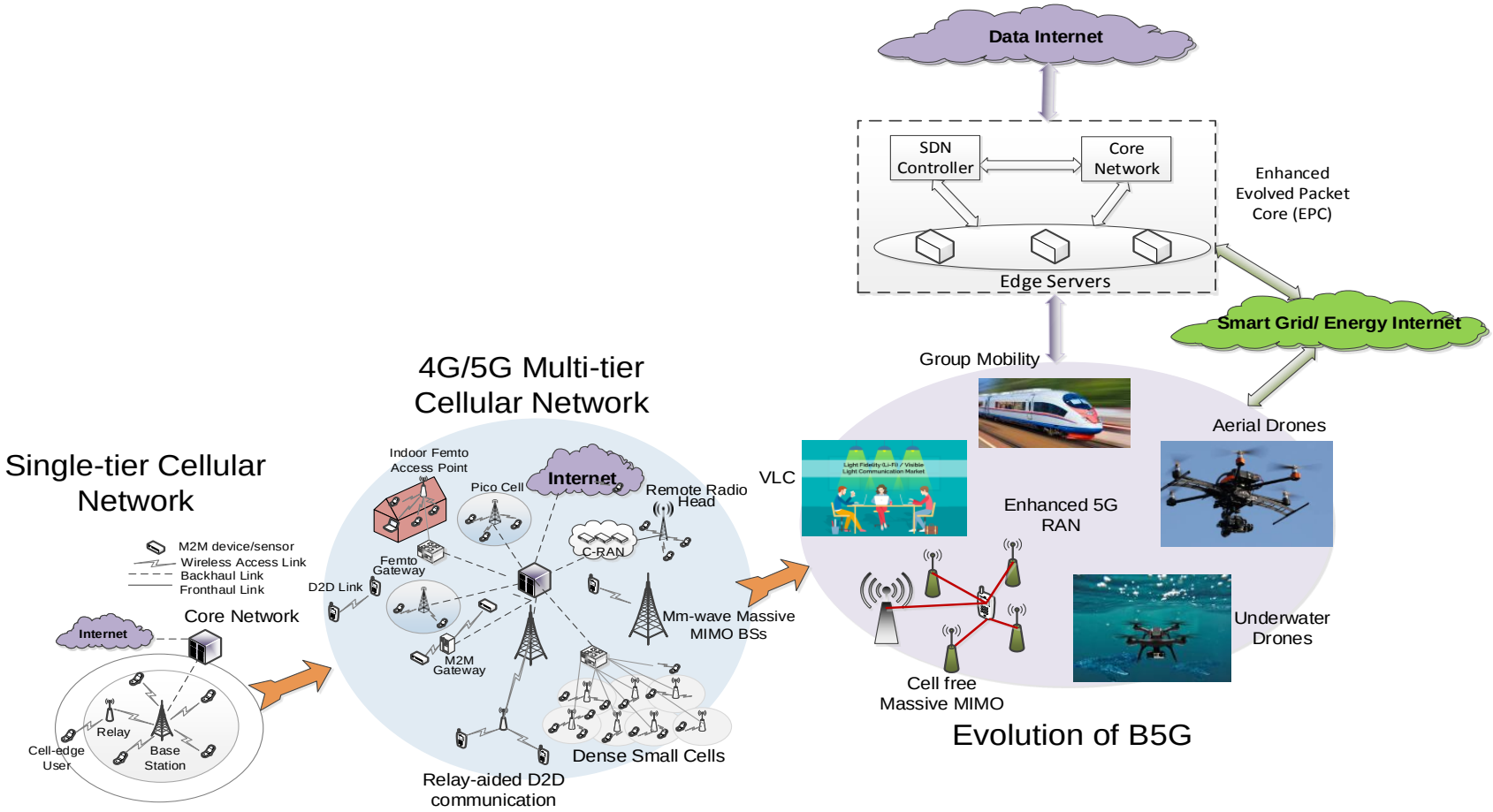
Ekram Hossain, *FIEEE*
University of Manitoba

March 1, 2021

Outline

- B5G Vision
- Multiple Access in B5G/6G
- DRL Techniques
- DRL-Based Design Example
- Conclusion

Evolution toward next generation wireless networks



Next generation wireless networks: Beyond 5G

- Internet-of-Everything
  - Converged terrestrial and non-terrestrial communication networks
  - Massive capacity, massive connectivity, energy efficiency
  - Programmable radio environment
  - Convergence of communications and computing
  - Virtualization/network slicing and "SDN"-ization of converged communication networks

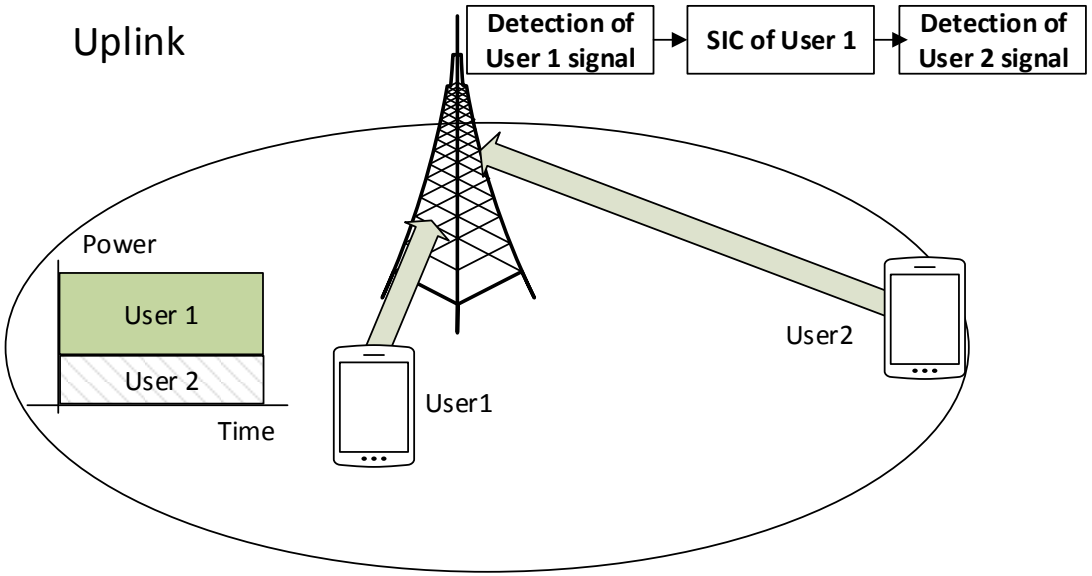
Next generation wireless networks: Beyond 5G

- Data rate $\sim$ 1 Tbps, compared to 10-20 Gbps in 5G, 1 Gbps in 4G
- Potential technologies:
  - THz and nm-wave communication (e.g. OWC, VLC, FSO)
  - Cell-free/Distributed “massive" antenna technology
  - Drone-aided communication
  - Intelligent reconfigurable surface (IRS)-aided communication

Potential multiple access technologies

- Classic orthogonal multiple access (OMA) schemes, e.g. TDMA, FDMA, CDMA, OFDMA will not be sufficient
- Power domain NOMA included in 3GPP (LTE-A), next general digital TV standard (ATSC 3.0), and 5G new radio (NR) standard
- Massive cellular non-orthogonal multiple access (m-NOMA)
  - Overlapping NOMA (O-NOMA)
  - Coordinated Multipoint NOMA (CoMP-NOMA)
- NOMA in cell-free/distributed “massive" antenna systems

NOMA in cellular uplink: An example



Achievable rate:

$$R_1 = B\log_2\left(1 + \frac{p_1\gamma_1}{p_2\gamma_2 + 1}\right)$$

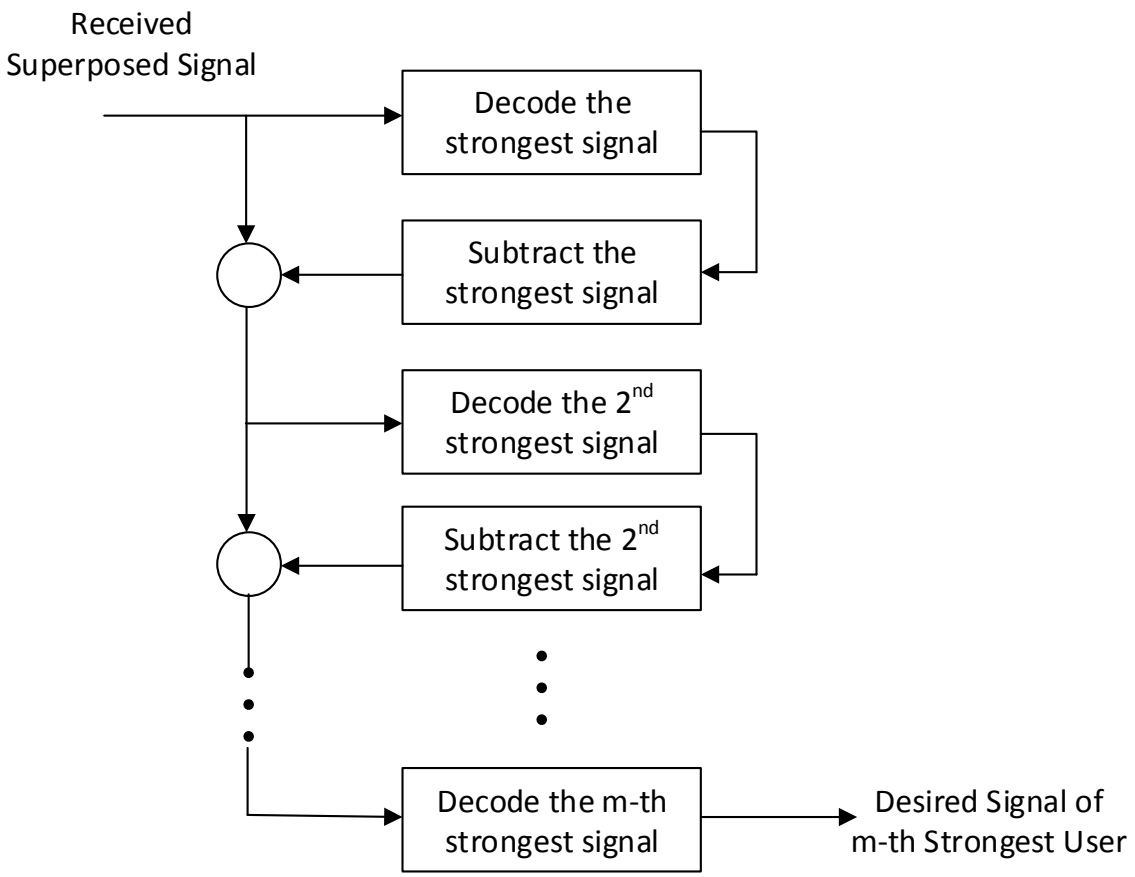
$$R_2 = B\log_2\left(1 + p_2\gamma_2\right)$$

Requirement for SIC:

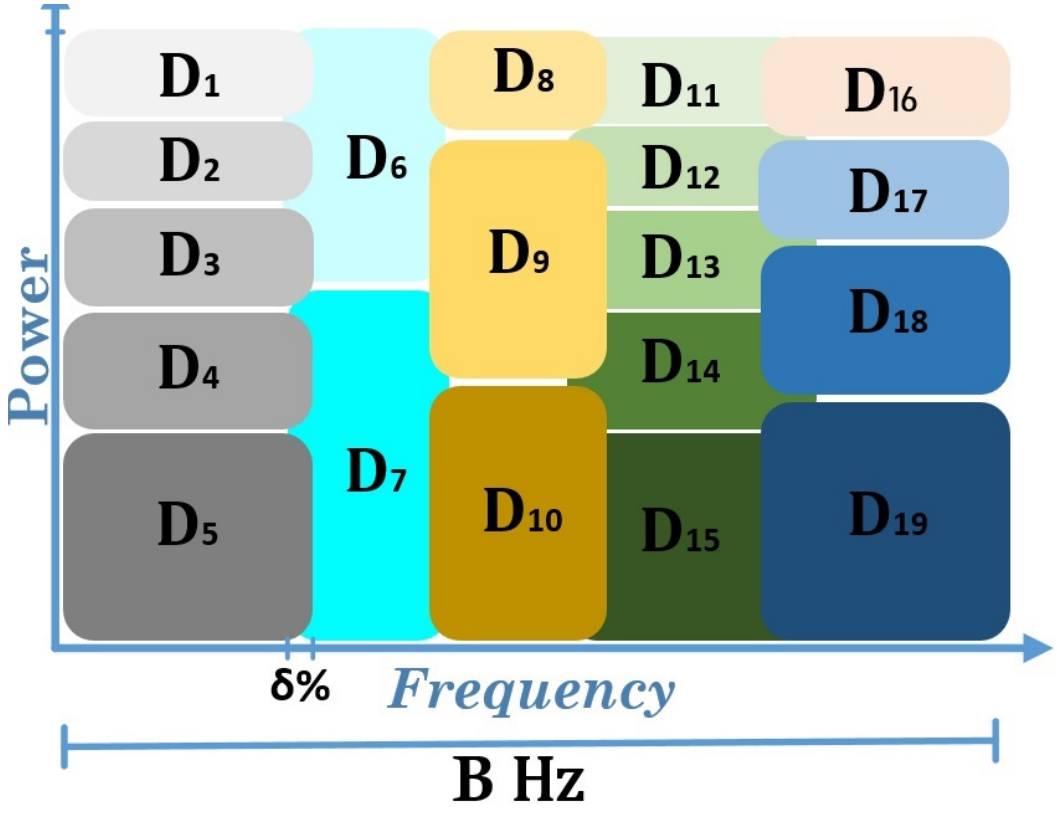
$$p_1\gamma_1 - p_2\gamma_2 \geq p_{tol}, \quad p_1 \leq p_t', \quad p_2 \leq p_t'$$

SIC in NOMA

- **Successive interference cancellation** (SIC) used at the receiver end to cancel intra-NOMA-user interference (INUI)



O-NOMA

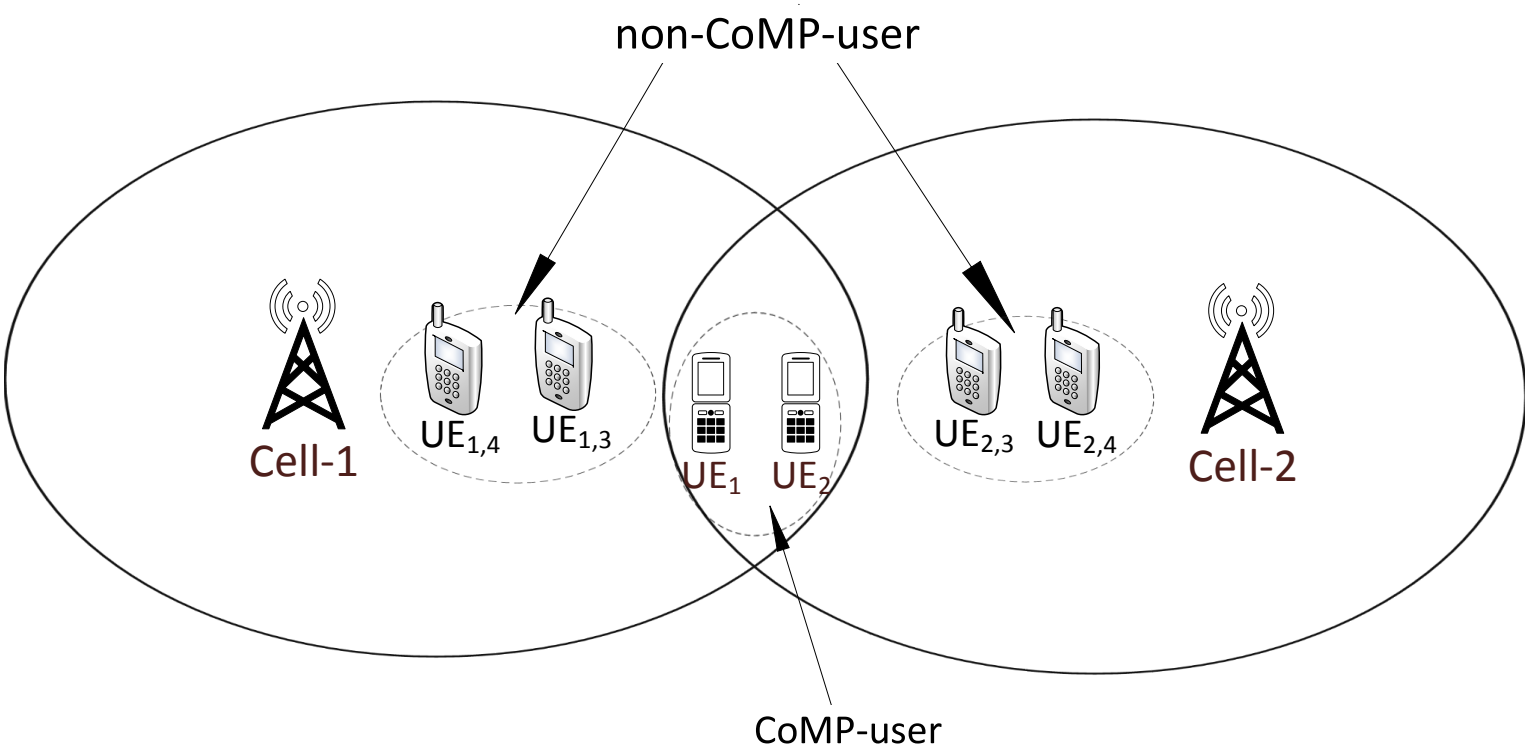


Y. Al-Eryani and E. Hossain, "The Delta-OMA (D-OMA) method for massive multiple access in 6G," *IEEE Vehicular Technology Magazine*, Special Issue on "6G: What is Next", vol. 14, no. 3, pp. 92-99, Sept. 2019.

O-NOMA

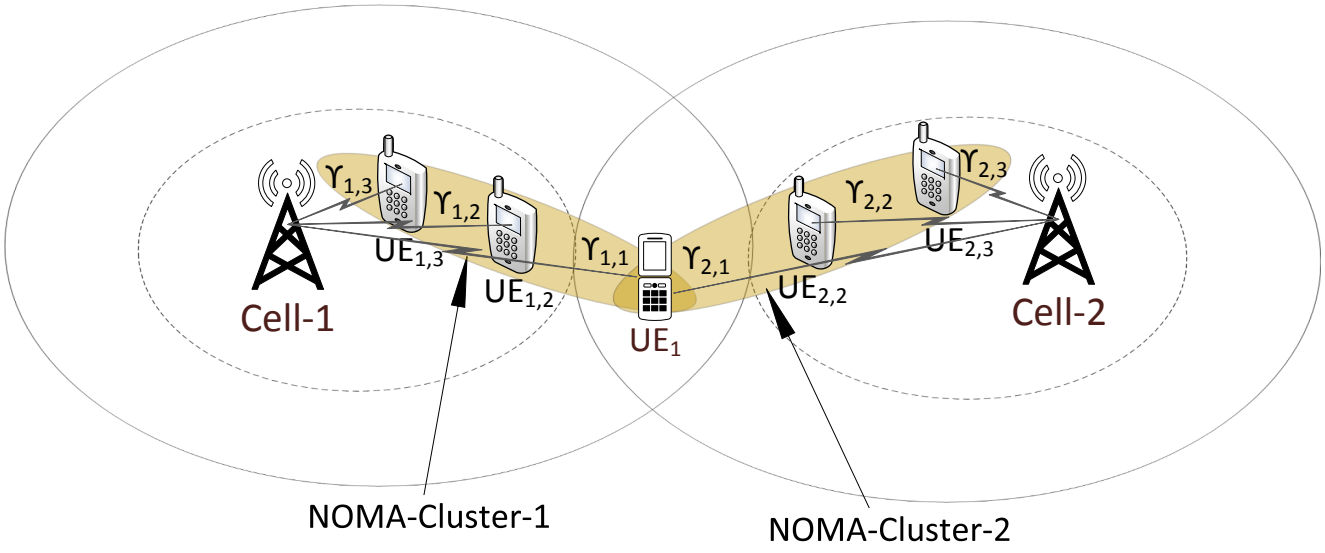
- Overlapping clusters $\longrightarrow$ less number of users per cluster for same spectral efficiency
- Trade-off between cluster overlapping and number of users per cluster
- For the same number of users per-cluster, overall spectral efficiency can be significantly enhanced by increasing the overlapping percentage among clusters.

CoMP (Coordinated Multi-Point)-NOMA



M. S. Ali, E. Hossain, and D. I. Kim, "Coordinated multi-point (CoMP) transmission in downlink multi-cell NOMA systems: Models and spectral efficiency performance," *IEEE Wireless Communications*, Special Issue on "Non-Orthogonal Multiple Access for 5G", vol. 25, no. 2, pp. 24-31, Apr. 2018..

CoMP-NOMA

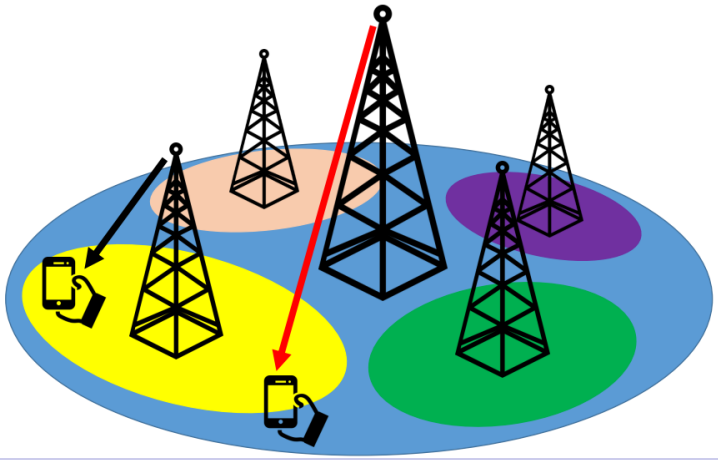


$$R_1 = B\log_2\left(1+\frac{\sum_{i=1}^{2} p_{i,1}\gamma_{i,1}}{\underbrace{\sum_{i=1}^{2}\sum_{j=2}^{3} p_{i,j}\gamma_{i,1}}_{\text{NOMA interference}}+1}\right), \quad R_{i,j} = B\log_2\left(1+\frac{p_{i,j}\gamma_{i,j}}{\underbrace{\sum_{k=j+1}^{3} p_{i,k}\gamma_{i,j}}_{\text{NOMA interference}} + \underbrace{\sum_{m=1,m\neq i}^{2}\sum_{l=2}^{3} p_{m,l}\gamma'_{i,j}}_{\text{Inter-cell interference}}+1}\right)$$

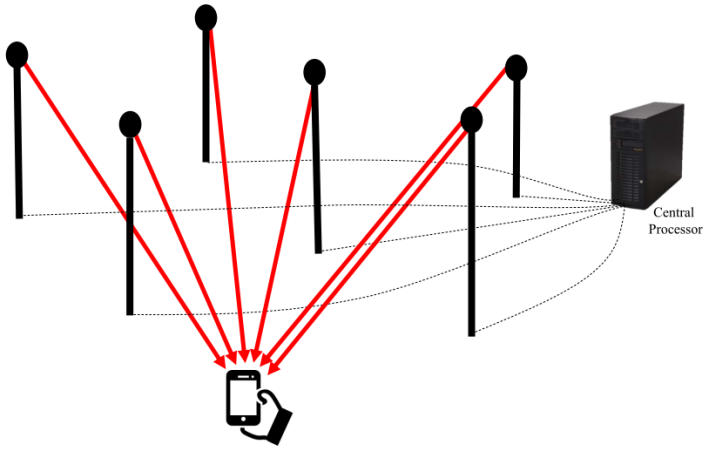
Cell-free NOMA

- Higher overlapping between NOMA clusters → higher interference levels.
- How to compensate for inter-cluster interference while maximizing spectral efficiency?
- Yes, through full BS/AP cooperation→ **cell-free network**
- Distributed single antenna system + Cooperative MIMO ⇒ Cell-free large MIMO

Cellular network

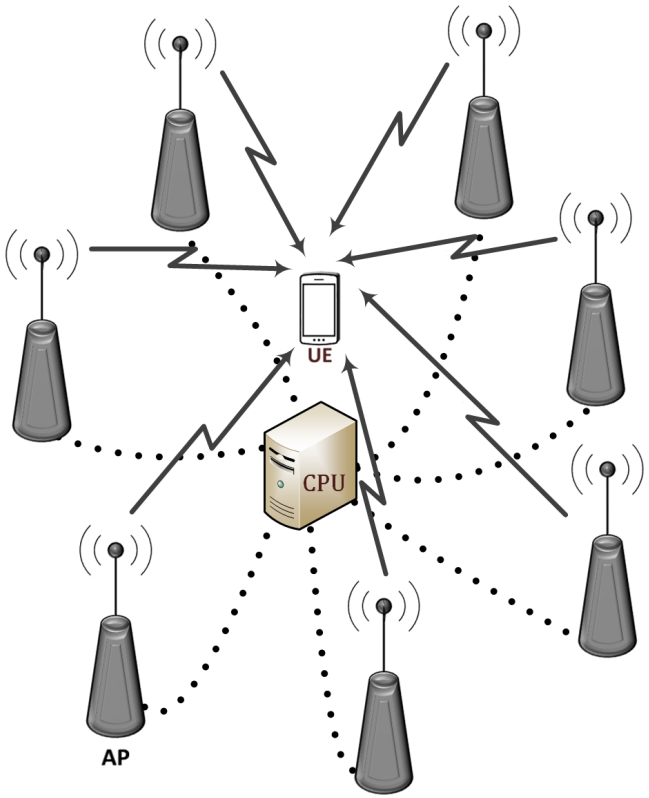


Cell-free network



SIC-enabled cell-free network

- All users (K users) access all spectrum using an optimized transmission power
- All APs (M APs) simultaneously serve all users
- Cell-free network increases per-user **rate** and network **coverage**.



SIC-enabled cell-free network

- Uplink achievable SINR for k-th user at the CPU:

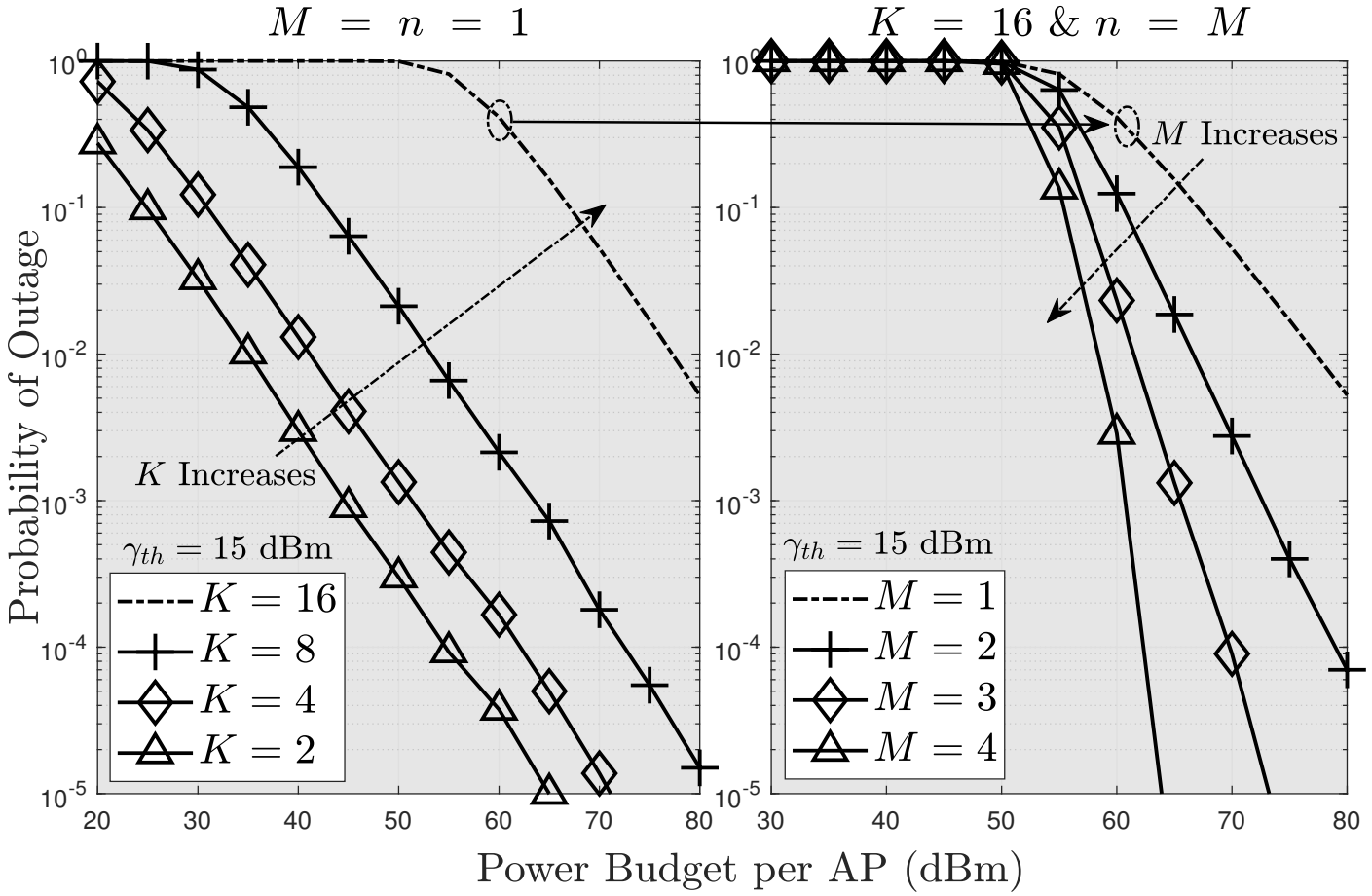
$$\gamma_k = \frac{\sum_{m=1}^{M} w_{mk}^2 |h_{mk}|^2}{\sum_{m=1}^{M} w_{mk}^2 \left(\sum_{l=1, l \neq k}^{K} |h_{ml}|^2 + N_m \right)}.$$

- If uplink channel gains are ordered such that:

$\sum_{m=1}^{M} |h_{m1}|^2 \leq \sum_{m=1}^{M} |h_{m2}|^2 \leq \cdots \leq \sum_{m=1}^{M} |h_{mK}|^2$, using SIC at user k, achievable SINR for user k:

$$\gamma_k = \frac{\sum_{m=1}^{M} w_{mk}^2 |h_{mk}|^2}{\sum_{m=1}^{M} w_{mk}^2 \left(\sum_{l=k+1}^{K} |h_{ml}|^2 + N_m \right)}.$$

Cell-free NOMA: Outage performance



Y. Al-Eryani, M. Akrout, and E. Hossain, "Multiple access in cell-free networks: Outage performance, dynamic clustering, and deep reinforcement learning-based design," *IEEE JSAC*, DOI: 10.1109/JSAC.2020.3018825.

General beamforming problem formulation

$$\mathbf{P}_1: \underset{\boldsymbol{W}\in[0\ 1]^{M\times K}}{\text{maximize}} \sum_{k=1}^{K} \log_2(1+\gamma_k)$$

$$\text{s.t. } \mathbf{C}_1: \left(w_{m\delta_l}^2 - \sum_{i=\delta_l+1}^{l} w_{mi}^2\right)\bar{\gamma}_{ml} \geq P_{\text{s}},$$

$$\mathbf{C}_2: w_k^2 \in [0 \quad 1],$$

$$\forall k, \forall\, \delta_l = 1,\ldots,l-1 \text{ , and } l = 2,\ldots,k,\ldots,K,$$

where $\bar{\gamma}_{ml} = p_l \sum_{m=1}^{M} |h_{ml}|^2$, and $\boldsymbol{W} \in [0\ 1]^{M\times K}$ is the overall beamforming matrix in which $\boldsymbol{w}_k = [w_{1k}\ldots w_{Mk}]$.

- A non-convex optimization problem
- Complexity increases as the number of APs and/or users increases.

ML for multiple access/resource management

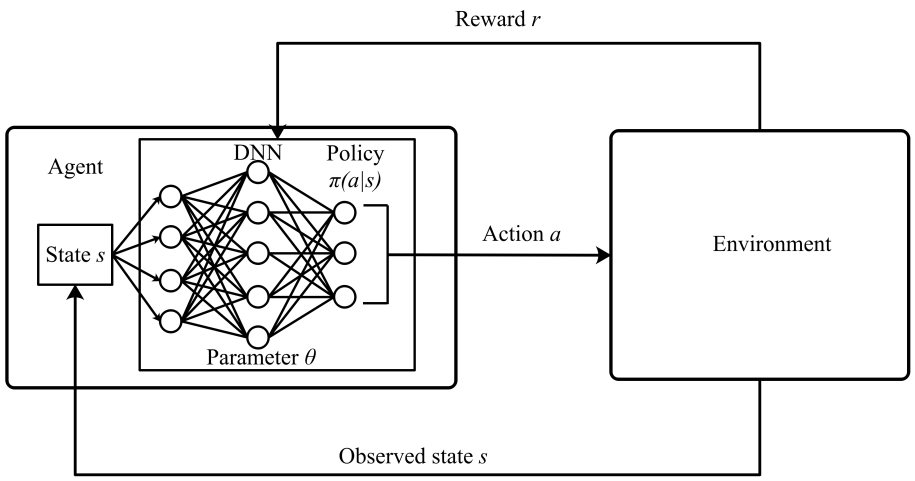
- When no mathematical model or efficient algorithm (modeling and/or algorithmic deficit) and task involves a function that maps well-defined inputs to well-defined outputs
- With ML, solutions are learned by exploiting the data samples (**data-driven supervised ML**)
- ML tools can be used to obtain practical solutions given the past optimal or near-optimal decisions.
- ML-based algorithms can be implemented online.
- No performance guarantee (suboptimal performance)
- When **data sets cannot be made available**, reinforcement learning (RL)/deep RL (DRL) techniques can be used.

RL

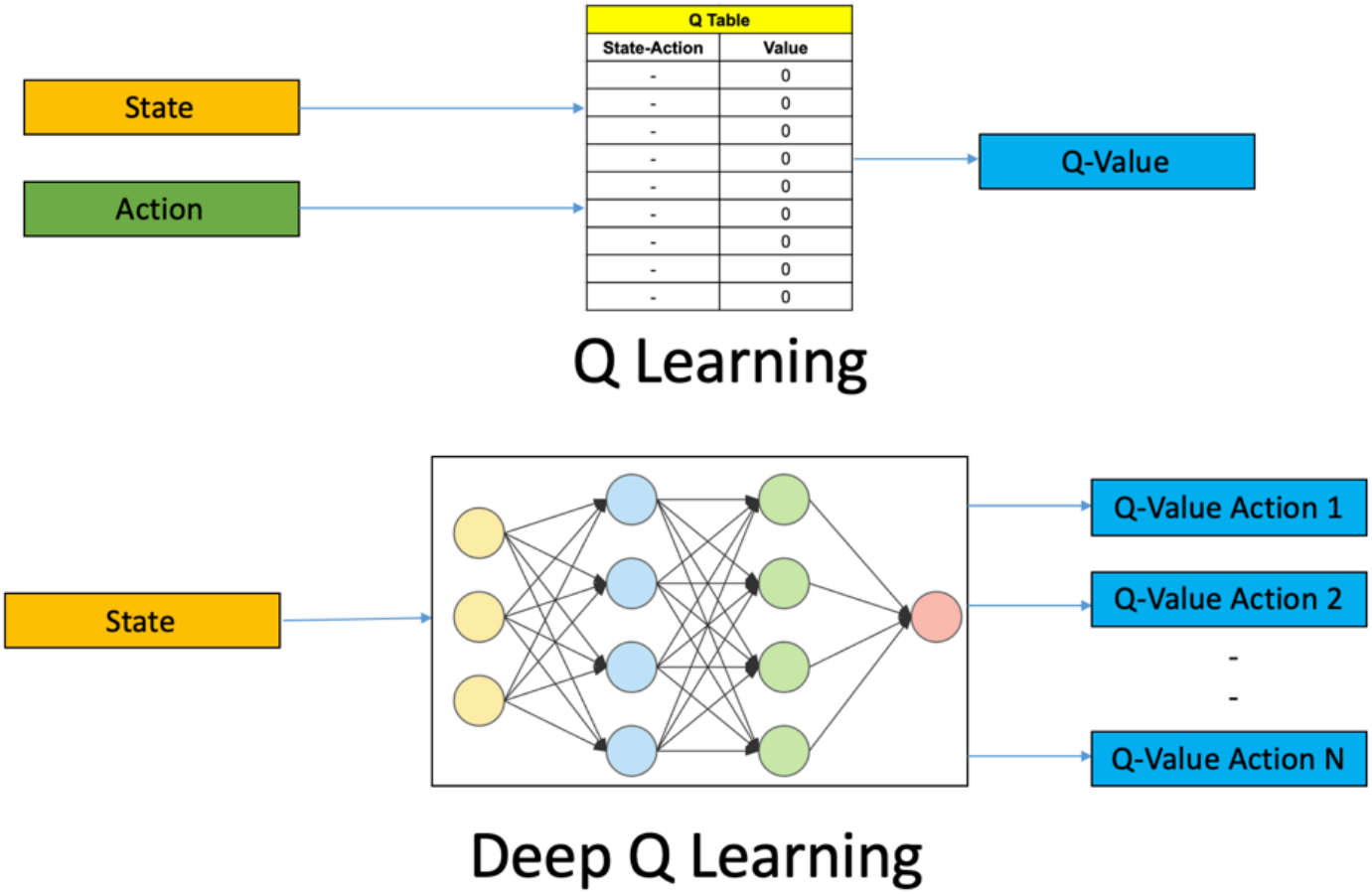
- In RL, an agent learns by interacting with the environment.
- At time step t, the agent receives state s_t from a state space $\mathcal{S}$ and takes an action a_t from an action space $\mathcal{A}$.
- The agent aims to maximize the expected cumulative reward through a series of actions.
- Agent follows a policy $\pi(a_t|s_t)$, i.e. a mapping from state s_t to action a_t, to choose the action.
- An action-value function $Q_\pi(s,a) = \mathbb{E}[r_t|s_t = s, a_t = a]$ is the expected return for selecting action a in state s and then follow policy π.
- Optimal action-value function $Q^*(s,a) = \max_\pi Q_\pi(s,a)$: maximum action value for state s and action a over all policies.
- **Optimal policy**: $\pi^*(s) = \arg\max_a Q^*(s,a)$

DRL

- In RL, Q-values are stored in tabular form.
- Traditional RL is limited to low dimensional state and action spaces.
- DRL: Combination of deep neural network (DNN) and RL
- In DRL, a DNN is used as an agent.



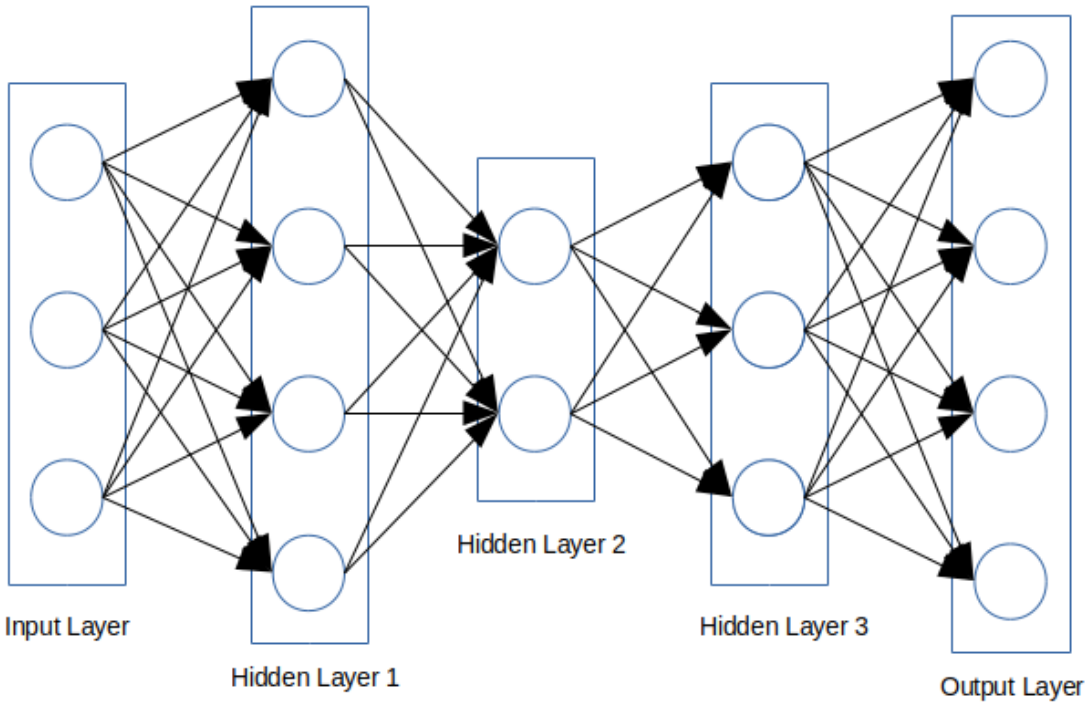
Deep Q-Learning (DQL)



https://www.analyticsvidhya.com/blog/2019/04/introduction-deep-q-learning-python/

Deep Neural Network (DNN)

- DNN: A deep learning architecture consisting of many hidden layers
- Defines a mapping $g(\mathbf{x}, \theta) : \mathbb{R}^n \rightarrow \mathbb{R}^k$ of an input vector $\mathbf{x} \in \mathbb{R}^n$ to an output vector $\mathbf{y} \in \mathbb{R}^k$.



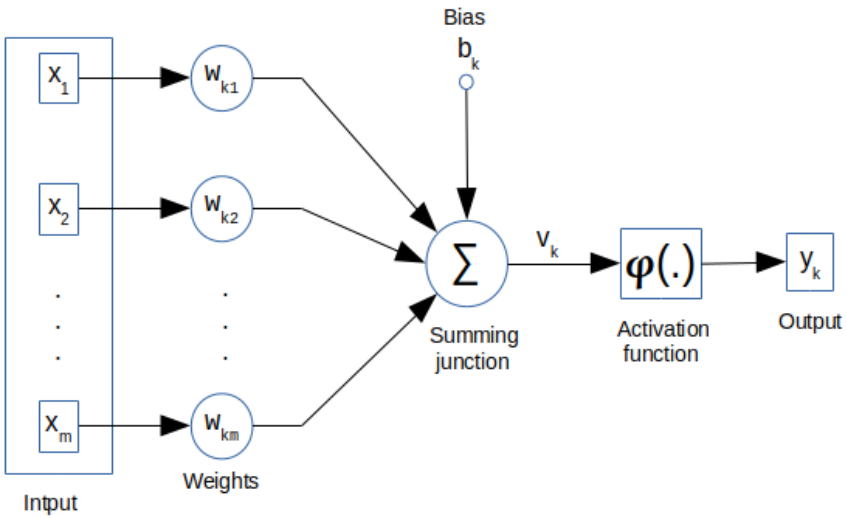
Non-linear model of a neuron

- Mathematically, for neuron k in a hidden layer

$$v_k = \sum_{j=1}^{m} w_{kj} x_j + b_k$$

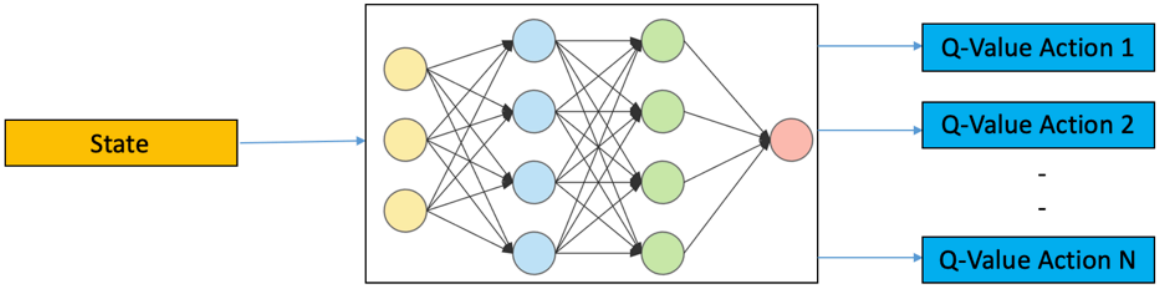
$$y_k = \varphi(v_k)$$

- Commonly used learning algorithm: Back propagation algo



DRL

- DRL algorithms can be classified into three types:
  - **Value-based methods** (e.g. deep Q-learning [DQL], state-action-reward-state-action [SARSA]), which only learn the value function to find a policy
  - **Policy-based methods**, which learn the policy directly
  - **Actor-critic methods** which are a hybrid of the value-based method for the critic and policy-based method for the actor



Deep Q Learning

DQL

- In DQL, a neural network is used to approximate the optimal action-value function:

$$Q(s, a; \theta) \approx Q^*(s, a)$$

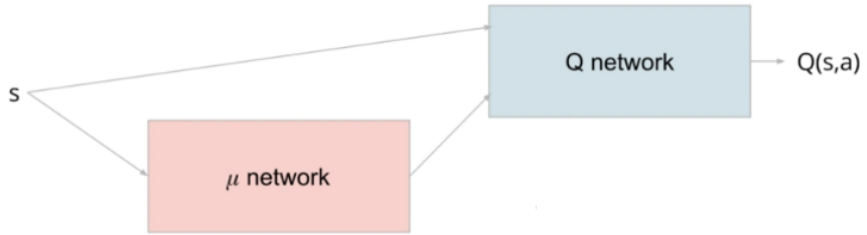
- $Q(s, a; \theta)$ is called the Q-network and θ is the parameter of neural network.
- **Optimal policy**: $\pi^*(s) = \arg\max_a Q^*(s, a)$

DDPG

- DDPG belongs to the class of actor-critic algorithms.
- Unlike the DQL policies which output the probability distribution $\pi(a|s)$ across a discrete action space $\mathcal{A}$, the policy network of DDPG directly maps states to actions.
- It concurrently learns a Q-function network approximation $Q(s, a|\theta^Q)$ called the *critic*, and a policy network approximation $\mu(s|\theta^\mu)$ called the *actor*.
- The policy is deterministic since it directly outputs the action.

DDPG

- To calculate the Q-value of a state, the actor output is fed into the Q-network to calculate the Q-value.
- *Target actor network* $\mu'(s|\theta^{\mu'})$: time-delayed copy of the actor network $\mu(s|\theta^{\mu})$.
- *Critic network* $Q(s,a|\theta^{Q})$: maps the state and the output action of $\mu(s|\theta^{\mu})$ to their corresponding Q-value.
- *Target critic network* $Q'(s,a|\theta^{Q'})$: time-delayed copy of the critic network $Q(s,a|\theta^{Q})$.



https://towardsdatascience.com/deep-deterministic-policy-gradient-ddpg-theory-and-implementation-747a3010e82f

DDPG: Actor (policy network) loss

- At every time step t, the *actor network* maximizes the function

$$J(\theta^{\mu}) = \mathbb{E}\Big[Q(s,a) \mid s = s_t, a = \pi(a|s_t)\Big]$$

and updates its weights θ by following the gradient of the above function (actor loss chain rule):

$$\nabla_{\theta^{\mu}} J(\theta^{\mu}) \approx \nabla_a Q(s,a)\, \nabla \mu(s|\theta^{\mu}).$$

DDPG: Actor (policy network) loss

- Since we update the policy in an off-policy way with batches of experience, we take the mean of the sum of gradients calculated from the mini-batch: $\nabla_{\theta^\mu} J(\theta^\mu) \approx \frac{1}{N}\sum_i \left[\nabla_a Q\left(s, a \mid \theta^Q\right)\Big|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu}\mu\left(s \mid \theta^\mu\right)\Big|_{s=s_i} \right]$
- This update rule represents the deterministic policy gradient (DPG) theorem.
- The term $\nabla_a Q(s,a)$ is obtained from the Q-network $Q(s,a|\theta^Q)$ (i.e. the *critic network*) by backpropagating its output w.r.t. the action input $\mu(s|\theta^\mu)$.

DDPG: Critic loss

- The critic network minimizes the following loss function:

$$\frac{1}{N}\sum_{i}\left(r_i+\gamma Q'\left(s_{i+1},\mu'(s_{i+1}|\theta^{\mu'})\middle|\theta^{Q'}\right)-Q(s_i,a_i)\right)^2.$$

- Similar to DQL, two tricks are employed to stabilize the training of the DDPG actor-critic architecture:
  - The experience replay buffer R to train the critic, and
  - The parameters $\theta^{Q'}$ and $\theta^{\mu'}$ in the two networks are updated by *Poylak averaging* with factor ρ as follows:

$$\theta^{Q'} \leftarrow \rho\,\theta^{Q}+(1-\rho)\,\theta^{Q'}$$

$$\theta^{\mu'} \leftarrow \rho\,\theta^{\mu}+(1-\rho)\,\theta^{\mu'}.$$

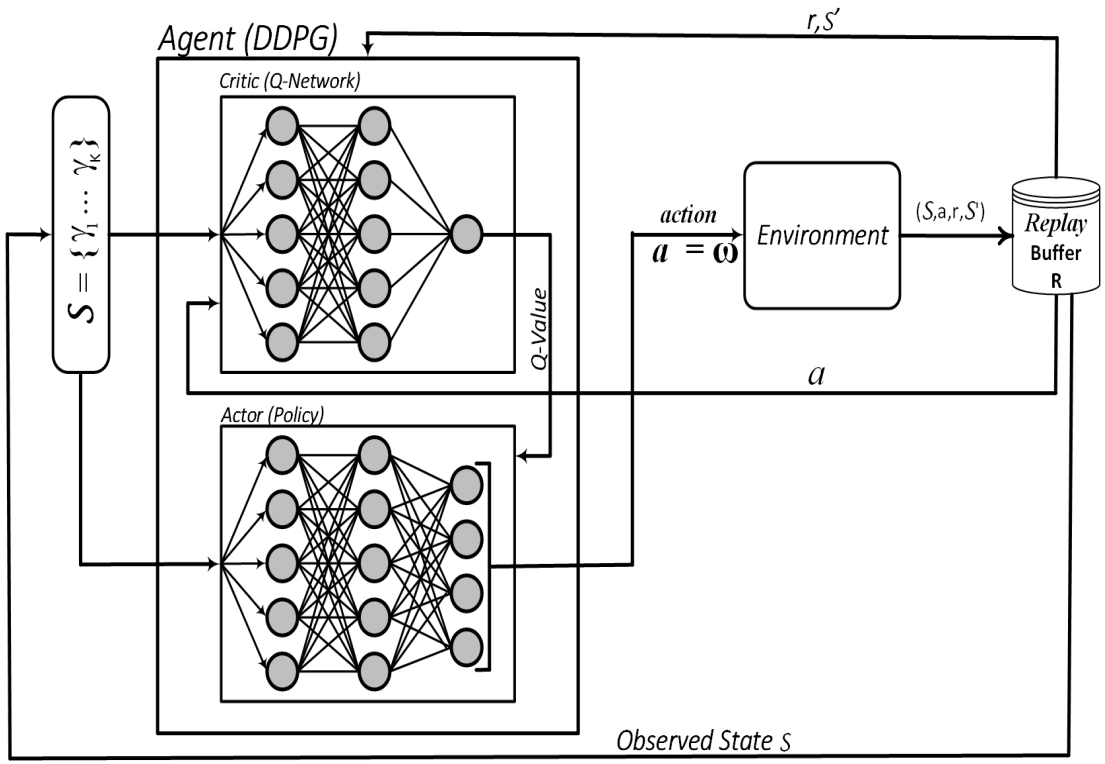
Example: DDPG-based beamforming in cell-free networks

- Use one DDPG (Deep Deterministic Policy Gradient) agent implemented in the CPU
- DDPG agent is composed of two networks: Actor network (also called policy network) and a critic network

Environment Variables	**System Equivalence**
State $s = \{s_1, ..., s_k\}$	User SINR: $\{\gamma_1, ..., \gamma_k\}$
Reward r	Sume-rate for all users: $\sum_{k=1}^{K} \log(1 + \gamma_k)$
Action a	Beamforming matrix: $\boldsymbol{W}$

DDPG-based beamforming

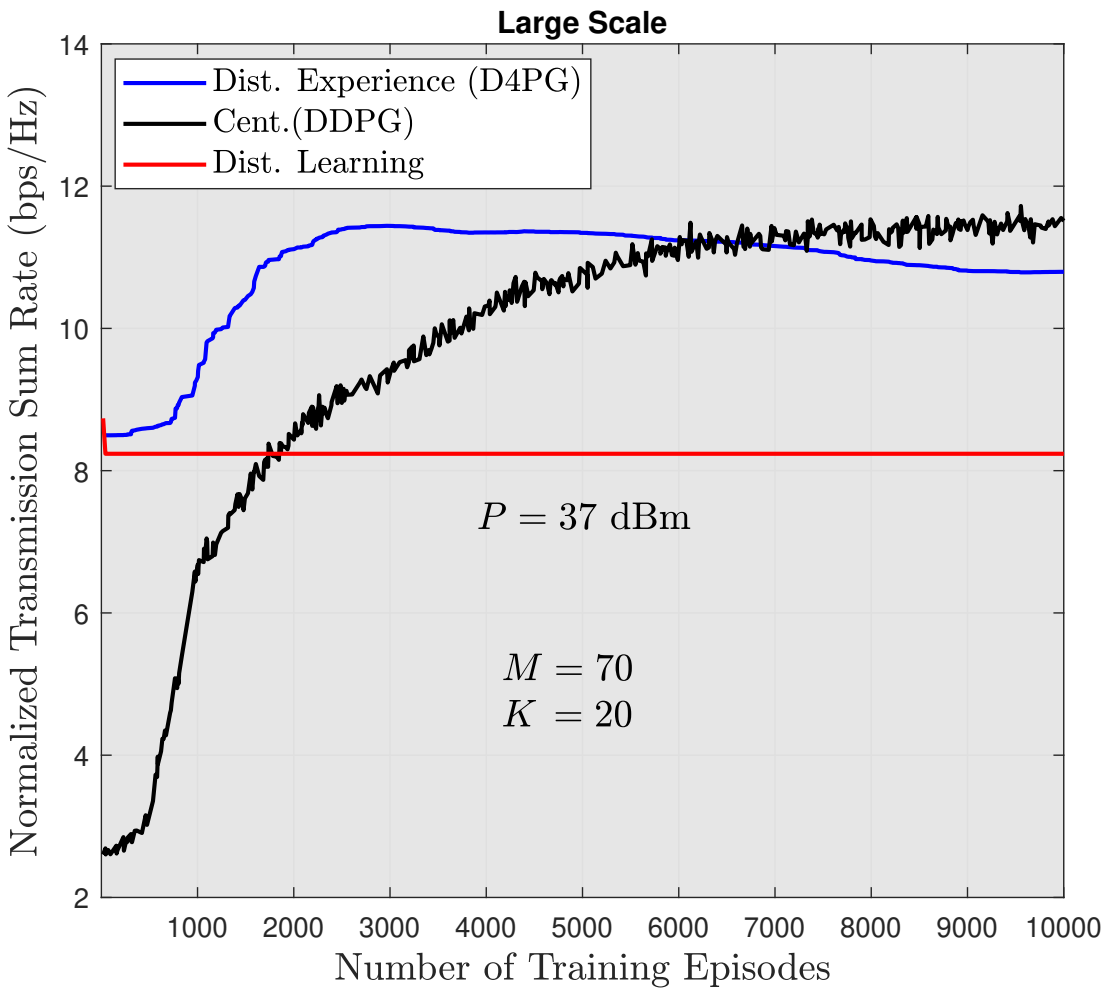
- Basically, the DDPG is composed of two steps: exploration (generating the replay table) and learning (training actor and critic networks from samples of the replay table).



Design parameters for the DRL model

- For the actor network: Input (state) = SINR, output = beamforming matrix
- *Actor network* $\mu(s|\theta^{\mu})$ maps the SINR values of users to the beamforming matrix $\boldsymbol{W}_k$. The output of the network is a flat list of all the elements $w_{ij} \in [0, 1]$.
- e.g. for a network with $K = 5$ users and $M = 8$ antennas, State = SINR $\in \mathbb{R}^{5\times 1}$ and Action = beamforming matrix $w \in [0, 1]^{40\times 1}$
- For the critic network: Input (State, Action)=(SINR, beamforming matrix), e.g. with $K = 5$ and $M = 8$, (State, Action) $\in \mathbb{R}^{(5+40)\times 1}$ and output=Q-value $\in \mathbb{R}$.

DRL-based beamforming: Performance results



Final remarks

- Cell-free NOMA for massive capacity and connectivity
- Integration with mmWave and nmWave communications (e.g. THz communications, visible light communications) technologies, as well as RIS and drone-aided/aerial communications
- ML techniques will be required for implementation-friendly multiple access/resource allocation in massive NOMA networks
- Deep reinforcement learning techniques are particularly suitable (e.g. to solve beamforming, pilot allocation problems)
  - Problem specific modeling
  - Leverage state-of-the-art DQL techniques
- Hardware design and implementation

Thank you!