

Distributional Deep Reinforcement Learning for Intelligent V2V Communication

Michael Ferko
Dept. of Electrical and Computer Engineering
University of Louisville
Louisville, USA
Email: Mike.W.Ferko@gmail.com

Abstract—A Deep Reinforcement Learning framework is introduced for V2V intelligent communication of messages. Optimality of the system is defined in terms of the Monte Carlo Method for a Markov Decision Process and expands into Deep Q-learning. Finally, the state-of-the-art C51 Rainbow is discussed, how it can be improved and how the DQN in V2V communication has developed.

Keywords—Intelligent Transport System, Vehicular Network, Cellular V2X, Deep Reinforcement Learning, Resource Allocation, Deep Q-learning Network, Distributional Bellman Operator, Machine Learning

I. INTRODUCTION (HEADING I)

The aspiring goal is that vehicles will communicate with anything, anytime and anywhere through worldwide wireless networking to improve road safety, situational awareness and travel comfort while reducing traffic congestion, air pollution and road infrastructure costs [1]. The central objective of this goal is a scalable and intelligent Vehicle-to-Everything (V2X) network capable of efficient information exchange between vehicles (Vehicle-to-vehicle (V2V) communication), roadside infrastructure (Vehicle-to-Infrastructure (V2I) communication) and everything in between (e.g. Infrastructure-to-Vehicle (I2V), Infrastructure-to-Infrastructure (I2I), etc.). The first stated purpose of the Intelligent Transport System (ITS) is road safety. A Safety message needs to be the first and primary message communicated among vehicles. Deciding which safety message is most important among multiple vehicles in potential danger needs to be selected quickly and intelligently.

Deep Reinforcement Learning (DRL) has recently been a promising area of Machine Learning research. Artificial Intelligence is a broad category which encapsulates Machine Learning and more specifically Deep Learning. Machine learning began with the idea to re-engineer the human brain. The neuron was a mathematical basis for the perceptron. The perceptron multiplies a weight to each input array. A perceptron node is where the weighted input array is summed together and passed through an activation function and the output of that perceptron node can be a 1 or a 0 (yes or no). A Neural Network (NN) can have many input arrays where each array is fully connected to a hidden layer of multiple perceptron nodes and a decision may be made based on the output layer. For example, a car can be an output classification by observing features like wheels, windows, doors and seats in the hidden layer and multiple pictures as input image arrays. A Deep Neural Network (DNN) has multiple hidden layers and the hyperparameter

tuning can be quite complex. NNs can be categorized as supervised, unsupervised, reinforcement, etc. DRL for V2V resource allocation specifically utilizes the Monte Carlo (discrete-time or hysteresis model) method from a game theory called the Markov Decision Process (MDP) for Deep Q-learning Networks (DQN) to perform data computational offloading tasks. DQNs are showing great promise in both online deployment where a DNN is actively learning to adjust the input parameters for a desired task performing algorithm like in robotics and traffic configuration and in offline deployment where a DNN learns from a large set of data and that interaction can be logged, reexamined and reused to make a more refined decision.

In this paper, we will use distributional deep reinforcement learning to handle resource allocation and the broadcast scheduling jointly. The main contributions of this paper are to introduce the reader to deployments of DQNs, the state of the art DQN, and the progress toward using DQNs to decide which resources need to be allocated for safety message broadcasting among a group of vehicles when unfamiliar dangerous situations arise ultimately providing a robust solution for improved road safety and one step closer to a V2X worldwide wireless network.

II. SYSTEM MODEL

In this section, the system model and problem of joint resource allocation and broadcast scheduling are presented.

As shown in Figure 1, the vehicular network contains M cellular users (CUEs) denoted by $\mathcal{M} = 1, 2, \dots, M$ demanding V2I links for high capacity communications. At the same time, there are K V2V users (VUEs) denoted by $\mathcal{K} = 1, 2, \dots, K$ demanding V2V links for broadcasting the safety messages to nearby vehicles, where each message is transmitted from one vehicle to a group of receivers within a surrounding area [2].

The interference in the V2I link is caused by background Additive White Gaussian Noise (AWGN) and the VUEs attempting to share the same sub-band being broadcasted to the infrastructure. A signal-to-interference-plus-noise ratio (SINR) of the interference to the V2I link can be expressed as

$$\gamma_m^c = \frac{P^c h_m}{\sigma^2 + \sum_{k \in \mathcal{K}} \rho_{m,k} P^d \tilde{h}_k}, \quad (1)$$

Where P^c and P^d are the transmission powers of CUE and VUE, respectively. σ^2 is the noise power, h_m is the channel

gain of the m^{th} CUE. \tilde{h}^k is the interference power gain of k^{th} VUE. $\rho_{m,k}$ is the spectrum allocation indicator where $\rho_{m,k} = 1$ when the k^{th} VUE reuses the spectrum of the m^{th} CUE and $\rho_{m,k} = 0$ otherwise. From the Equation (1) SINR the capacity of the m^{th} CUE can be expressed as

$$C_m^c = W \cdot \log(1 + \gamma_m), \quad (2)$$

where W is the bandwidth, and this is a base 10 logarithm.

Similarly, for the j^{th} receiver of k^{th} VUE the SINR may be expressed as

$$\gamma_{k,j}^d = \frac{P^c h_m}{\sigma^2 + G_c + G_d}, \quad (3)$$

which is an intuitively compressed form. Where G_c is expressed as

$$G_c = \sum_{m \in \mathcal{M}} \rho_{m,k} P^c \tilde{g}_{m,k}, \quad (4)$$

and G_d is expressed as

$$G_d = \sum_{m \in \mathcal{M}} \sum_{k' \in \mathcal{K}, k' \neq k} \rho_{m,k} \rho_{m,k'} P^d \tilde{g}_{k',k,j}^d, \quad (5)$$

where g_k is the power gain of the k^{th} VUE, $\tilde{g}_{k,m}$ is the interference power gain of m^{th} CUE, and $\tilde{g}_{k',k,j}^d$ is the interference power gain of k'^{th} VUE. From the Equation (2) the capacity of the k^{th} VUE can be expressed as

$$C_{k,j}^d = W \cdot \log(1 + \gamma_{k,j}^d). \quad (6)$$

To increase reliability of the V2V broadcast communication, some vehicles need to rebroadcast a message so that more vehicles can get the same message within a latency constraint. When messages are rebroadcasted too many times a broadcast storm can occur where a massive amount of broadcast packets overwhelm the receivers. To reduce the broadcast storm, select the appropriate message for rebroadcast, and fulfill the latency constraint vehicles need to observe the environment and make a decision that will minimize V2I interference and meet the V2V latency constraint.

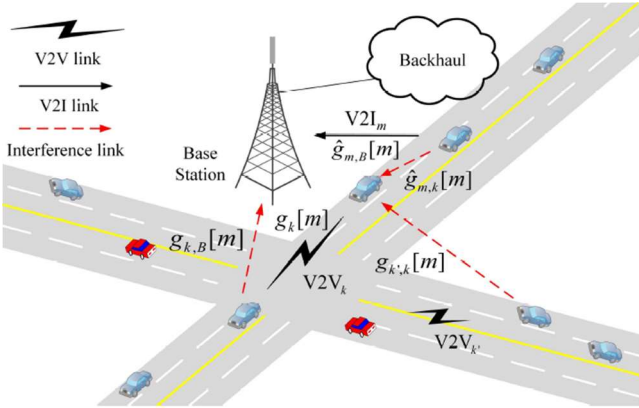


Figure 1: Structure of Vehicular Communication Network

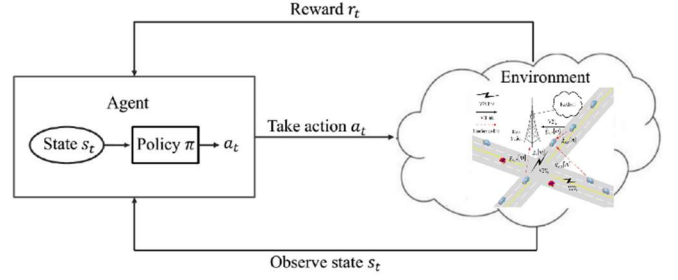


Figure 2: DRL for V2V Communications

III. DISTRIBUTIONAL RESOURCE ALLOCATION

In this section the framework for reinforcement learning resource allocation is built. The Monte Carlo discrete-time system, Markov decision Process, and Q-learning will be introduced.

A. Game Theory in Reinforcement Learning

The Markov Decision process is a 5-tuple (state, action, transition, reward, discount factor) game theory which when applied to DRL can find an optimal policy for broadcasting V2V messages. The agent (or vehicle) observes some state, s_t from a set of possible states in a state space $S = \{s_t, s_{t+1} \dots s_{N_{RB}+1}\}$. Performs some action a_t from a set of possible actions in the action space $A = \{a_t, a_{t+1} \dots a_{N_{RB}+1}\}$. The agent will then transition from state s_t to state s_{t+1} and a reward r_t will be given to the agent for taking action a_t . The conditional probability of transition matrix can be expressed as

$$\mathbf{P} = \{p(s_{t+1}, r_t | s_t, a_t)\} \quad (7)$$

And the expected cumulative discounted reward for all transitions can be expressed as

$$G_t = \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \right], 0 \leq \gamma \leq 1 \quad (8)$$

where the γ^k is the discount factor for the k^{th} observed state action pair.

For the case of distributional resource allocation a MDP state observed by each vehicle for each received message consists of several parts: the instant channel interference power to the link, I_{t-1} , the channel information of the V2I link, e.g. from the V2V transmitter to the base station, H_t , the selection of sub-channel neighbors in the previous time slot, N_{t-1} , the number of received times, O_t , the minimum distance to the vehicles that have broadcast the message, D_t , and the remaining time to meet the V2V latency constraints, U_t . To summarize, the state can be expressed as $s_t = [I_{t-1}, H_t, N_{t-1}, O_t, D_t, U_t]$.

Each action corresponds to deciding which messages should be broadcast and cognitively choosing sub-bands that are not in use. The action space is of length $N_{RB} + 1$ where N_{RB} is the number of resource blocks in an Orthogonal Frequency-Division Multiplexing (OFDM) logarithmic power spectrum.

Using a MDP will allow selection of messages and frequency bands that will minimize V2I interference while

meeting the V2V latency constraints. A reward function consists of three parts: capacity of V2I links, C_m^c , the capacity of V2V links, $C_{k,j}^d$, and the latency condition.

To reduce the effects of a broadcast storm, only capacities of receivers that have not received the message are included in the calculation of the reward function. This reward function can be expressed as

$$r_t = \lambda_c \sum_{m \in \mathcal{M}} C_m^c + \lambda_d \sum_{k \in \mathcal{K}, j \notin E\{k\}} C_{k,j}^d - \lambda_p (T_0 - U_t), \quad (9)$$

Where T_0 is the constraint time, and λ_c , λ_d and λ_p are the weights of the V2I links, V2V links and latency condition respectively, and $E\{k\}$ represents the set of expected receiver antenna that have received the transmitted message to minimize capacities of message signals that have already been received.

B. Monte Carlo Method for MDP and the Q-function

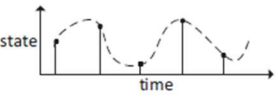
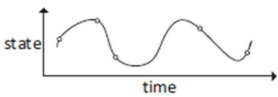
MDP	Semi-MDP
	
Discrete time	Continuous time
Homogeneous discount	Interval-dependent discount

Figure 3: Types of MDP

A Monte Carlo discrete-time MDP observes each state action pair iteratively creating a hysteresis of the probabilities of transition matrix. In Figure 3 we see the Monte Carlo method on the left creates an n-point cumulative distribution (CDF). However, since a cumulative distribution function cannot be numerically computed, the $Q(\cdot)$ function may be used to calculate the conditional probability of transition matrix. In many papers the $Q(\cdot)$ function is referred to as the distributional Bellman Operator. This name refers to the $Q(\cdot)$ function being used to approximate the CDF to evaluate the conditional probability of transition from observation of state s_t to the next state s_{t+1} and the bellman optimality equation for a MDP. The 1957 Bellman optimality proves that an optimal policy, π^* , (to perform an action) will be found when an optimal action value function, Q^* , is found. To find this optimal action value $Q(\cdot)$ function a Q-table needs to be updated iteratively and expressed in terms of the returned expected cumulative discounted rewards, G_t . The optimal policy may be expressed as

$$\pi^* = \arg \max_{\pi} \mathbb{E} \left[\sum_{t=0}^H R_t(s_t, A_t, s_{t+1}) | \pi \right] \quad (10)$$

And the optimal action value $Q(\cdot)$ function may be expressed as

$$Q^*(s, a) = \mathbb{E} \left[r + \gamma \max_{a'} Q^*(s', a') | s, a \right] \quad (11)$$

C. Q-learning

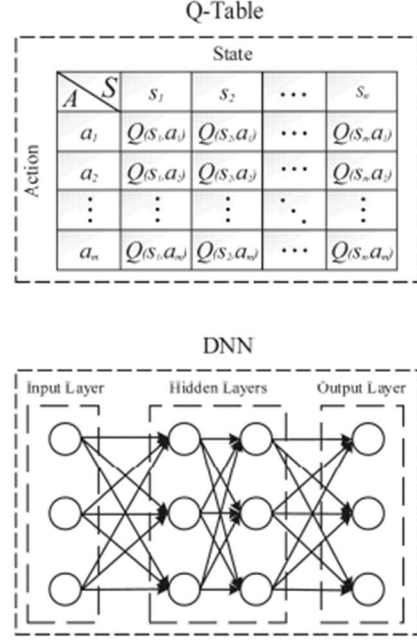


Figure 4: Q-Table and DNN

Q-learning utilizes the Monte Carlo Method for the MDP to converge Q values to the optimal Q-value Q^* with probability of 1 given that each action has been tried an infinite number of times under each state. The iterative process of updating Q-values may be expressed as

$$Q_{new}(s_t, a_t) = Q_{old}(s_t, a_t) + \alpha [r_{t+1} + \gamma \max_{s \in S} Q_{old}(s, a_t) - Q_{old}(s_t, a_t)]. \quad (12)$$

The action that maximizes the long-term accumulated returned rewards, G_t , may be expressed as

$$a_t = \arg \max_{a \in A} Q(s_t, a). \quad (13)$$

This means that the vehicle (agent) will Q-learn the correct action which is selection of messages and frequency bands that will minimize V2I interference while meeting the V2V latency constraints. This Q-learning will allow safety messages to be selected for broadcast, improving road safety, and ultimately enabling worldwide wireless communication through the V2X efficient exchange of information component.

D. Deep Q-learning

Q-learning is an effective solution, but the larger a state-action space becomes, the more time it will take to find an optimal Q-value. Since π^* is a mapping from states in S to probabilities of selecting each action in A the DNN can address this sophisticated mapping of channel information to desired output [2] which is message and frequency sub-band selection. A Deep Q-learning Network (DQN) updates the weights iteratively to minimize the L2 Loss function expressed as

$$Loss(\lambda) = |r_t + \max_{a \in A} Q(s_t, a, \lambda) - Q(s_t, a_t, \lambda)|^2. \quad (14)$$

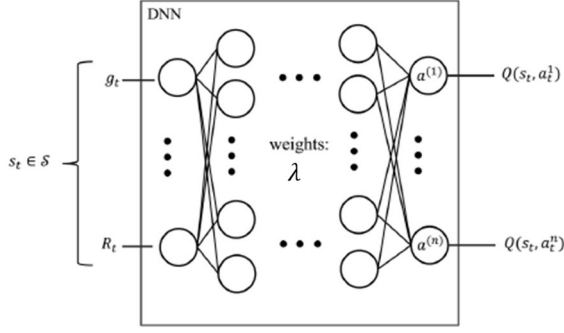


Figure 5: Deep Q-learning Network Architecture

The DQNs as shown in Figure 5 are trained with a large amount of simulated data, which are generated from interactions of agents and an environment simulator. Each sample contains s_t , s_{t+1} , a_t and r_t observed from randomly dropped vehicles. Each vehicle has a VUE and CUE and associated channels. During the training stage many papers will refer to the Q-learning with experience replay setup [3] to reduce any unintentional correlation of generated data.

IV. ADVANCEMENTS IN DQNS FOR V2V BROADCASTING

In this section we look at the state-of-the-art DQN and a few advancements for the application of DQNs to fulfill V2V broadcasting.

A. State-of-the-art DQN C51 Rainbow

The state-of-the-art benchmark DQN is the C51 Rainbow which is the combination of seven published baseline DQN algorithms [4]. A lot of the time evaluation of a DQN algorithm is done by comparison of median human-normalized performance across 57 Atari 2600 Learning Environment (ALE) games. The performance of these DQNs may be seen in Figure 6.

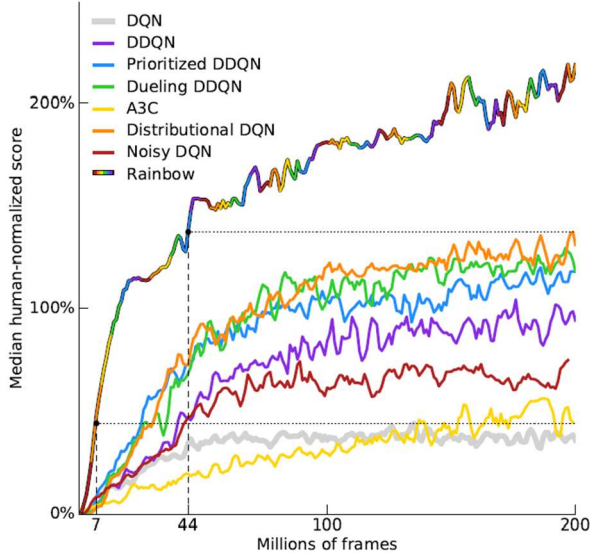


Figure 6: Median human-normalized performance across 57 Atari games for 7 published DQN algorithms

From Figure 6 shows Rainbow matches the original DQN's best performance after 7M frames, surpass any baseline within 44M frames, and reaches substantially improved final performance. Curves are smoothed with a moving average over 5 points. Also an ablation study was performed to show the contribution of each DQN algorithm to the performance of rainbow.

B. Spectrum Sharing in V2V Multi-Agent DRL

Spectrum Sharing in Vehicular Networks Based on Multi-Agent Reinforcement Learning [8] demonstrated that through a centralized training stage (offline deployment) and a distributed implementation stage (online deployment) the proposed resource sharing scheme is effective in encouraging cooperation among V2V links to improve system level performance. The paper suggested future work should include an analysis of robustness to understand when the trained Q-networks need to be updated and how to efficiently perform such updates.

C. DRL for Channel and Signal Detection in OFDM Systems

Power of Deep Learning for Channel Estimation and signal Detection in OFDM Systems [16] viewed OFDM and wireless channels as black boxes to see the robustness of DNNs to a few parameters. This paper showed that DNNs are robust to the number of pilot symbols, a Cyclic Prefix (CP) is necessary to convert the linear convolution of the physical channel into circular convolution and mitigate ISI, and Deep learning method is more robust to nonlinear clipping noise. Each of these parameters cause a slight divergence of the bit error rate (BER) at 15dB. The combination of all the effects causes a very large divergence as shown in Figure 7.

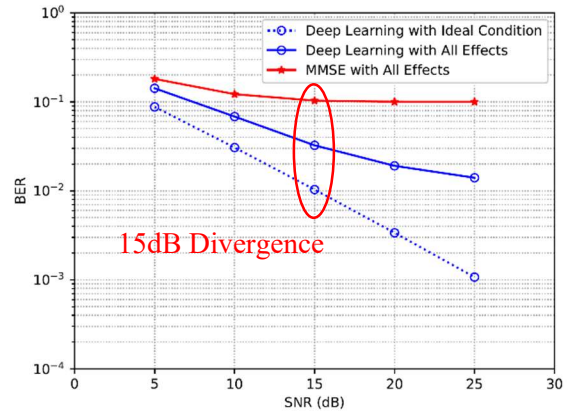


Figure 7: BER Curve when combining all adversities The BER can be expressed as

$$\text{BER} = \frac{1}{2} \text{erfc}(\sqrt{E_b/N_0}) \quad (15)$$

Where the normalized signal-to-noise ratio is E_b/N_0 .

D. Distributional RL with Quantile Regression

A recent advancement by the google deep mind team in their paper Distributional Reinforcement Learning with Quantile Regression [7] shows how conditional quantile regression was able to improve the performance results over the C51 Rainbow

DQN. This DQN expands and contracts quantile upper and lower bounds to cover a true range of returns, G_t , that accurately describes the distribution. The results are shown in Figure 8 use the same framework for the C51 state-of-the-art that most papers on DQNs will reference with better results.

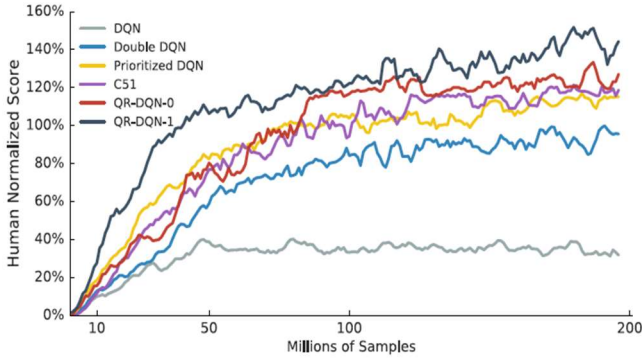


Figure 8: Comparison of Quantile Regression-DQN to C51

V. CONCLUSIONS

In this paper the Monte Carlo Method for the MDP Deep Q-learning Networks was discussed for deciding which frequency sub-bands and messages need to be selected. It was determined that the Optimal policy, π^* , for mapping states to actions is found by converging the Q action value closer and closer to an optimal Q^* approximating a probability distribution of 1 given all actions and states (Bellman's 1957 optimality Equation). The optimal approximations for action mapping can be found by maximizing the expected cumulative discounted returned rewards, G_t , and minimizing the L2 loss function. All of which was defined within the framework of weighted capacity of V2V links, V2I links and the latency condition.

The State-of-the-art C51 Rainbow DQN was introduced as a vital benchmark as well as expanded into the Quantile-Regression-DQN showing a refinement for distributional Bellman Optimality conditions. Recent advancements in spectrum sharing and message selection encourages cooperative decision-making patterns for V2V communication and robustness of some characteristic OFDM parameters was slightly discussed.

Further research should include dynamic programming (DP). Currently, the proposed scheme wastes time learning and does not reuse rewards upon redeployment. With DP the system should be able to choose an available model or resort to a self-learning DQN to reduce time between decisions and remember rewards from previous DQN deployment.

There are currently many autonomous vehicles from companies e.g. Waymo that drive in California collecting massive amounts of data that can be used to refine these Deep Reinforcement Learning models. These autonomous systems are on the cusp of realization and it is exciting to dream of the possibilities the future of technology has to offer.

VI. ACKNOWLEDGMENT

Thank you to Dr. Jacek Zurada for writing a clearly defined book on an Introduction to Artificial Neural Systems [17] and

teaching such a great first ever course at the University of Louisville just before the Covid-19 pandemic hit the United States. The course and book were and will be a vital component to understanding Deep Neural Networks for me and future generations.

Thank you Dr. Hongxiang Li for a modern approach to digital communications in the form of OFDM and MIMO systems. And for encouraging independent research into modern communication topics for himself and for his students.

VII. REFERENCES

- [1] Noor-A-Rahim, Md., et al. "A Survey on Resource Allocation in Vehicular Networks." 24 Aug. 2020.
- [2] Ye, Hao, and Geoffrey Ye Li. "Deep Reinforcement Learning Based Distributed Resource Allocation for V2V Broadcasting." *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*, 2018, doi:10.1109/iwcmc.2018.8450518.
- [3] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. H. I. Antonoglou, D. Wierstra, and M. A. Riedmiller, "Human-level control through deep reinforcement learning,"
- [4] Hessel, Matteo, et al. "Rainbow: Combining Improvements in Deep Reinforcement Learning." 6 Oct. 2017. *Nature* vol. 518, no. 7540, pp. 529–533, Feb. 2015 *Nature*, vol. 518, pp. 529–533, 2015.
- [5] Aeberhard, Michael, et al. "Track-to-Track Fusion with Asynchronous Sensors and out-of-Sequence Tracks Using Information Matrix Fusion for Advanced Driver Assistance Systems." *2012 IEEE Intelligent Vehicles Symposium*, 2012, doi:10.1109/ivs.2012.6232115.
- [6] Bellemare, Marc G., et al. "A Distributional Perspective on Reinforcement Learning." 21 July 2017.
- [7] Dabney, Will, et al. "Distributional Reinforcement Learning with Quantile Regression." 27 Oct. 2017.
- [8] Liang, Le, et al. "Spectrum Sharing in Vehicular Networks Based on Multi-Agent Reinforcement Learning." *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, 2019, pp. 2282–2292., doi:10.1109/jsac.2019.2933962.
- [9] Min, Kyushik, et al. "Deep Q Learning Based High Level Driving Policy Determination." *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, doi:10.1109/ivs.2018.8500645.
- [10] Noor-A-Rahim, Md., et al. "A Survey on Resource Allocation in Vehicular Networks." 24 Aug. 2020.
- [11] Pritzel, Alexander, et al. "Neural Episodic Control." 6 Mar. 2017.
- [12] Qin, Zhijin, et al. "20 Years of Evolution From Cognitive to Intelligent Communications." *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 1, 2020, pp. 6–20., doi:10.1109/tccn.2019.2949279.
- [13] Yavas, Ugur, and Nazm Kemal Ure. "A New Approach for Tactical Decision Making in Lane Changing: Sample Efficient Deep Q Learning with a Safety Feedback Reward."
- [14] Ye, Hao, and Geoffrey Ye Li. "Deep Reinforcement Learning Based Distributed Resource Allocation for V2V Broadcasting." *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*, 2018, doi:10.1109/iwcmc.2018.8450518.
- [15] Ye, Hao, et al. "MACHINE LEARNING FOR VEHICULAR NETWORKS." *IEEE Vehicular Technology Magazine*, June 2018, pp. 2–9.
- [16] Ye, Hao, et al. "Power of Deep Learning for Channel Estimation and Signal Detection in OFDM Systems." *IEEE Wireless Communications Letters*, vol. 7, no. 1, 2018, pp. 114–117., doi:10.1109/lwc.2017.2757490.
- [17] Zurada, Jacek M. *Introduction to Artificial Neural Systems*. West Publishing Company, 1992