# CSC 5930 Final Project Proposal: Application of Deep Q-Learning in Approximate Computing Paradigm

Rahul Thapa
*Undergraduate Researcher, Department of Computer Science*
*Villanova University*
Villanova, PA
rthapa@villanova.edu

## I. Introduction

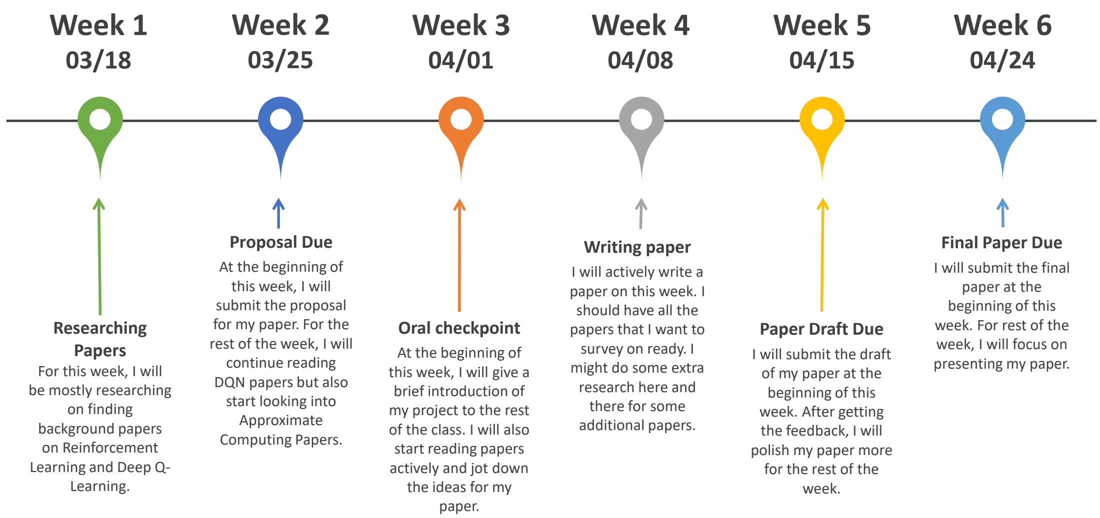
For the final project, I am writing a survey paper including an extensive study of the Deep Q-Learning (DQN) framework and approximate computing paradigm. I will present a study on how one can use DQN in applications such as edge detection using a Sobel filter to find the combination of approximate adders and multipliers that maximizes the energy efficiency while staying under a quality constraint.

## II. Project Description

Approximate computing trades of accuracy for energy efficiency for a given application which varies according to the uses in a certain field. For example in [1], the authors explore various configurations of multiple approximate resources and select the one that maximizes energy consumption within a quality constraint. They evaluate their model using several machine learning applications such as linear regression and SVM. This idea of trading off accuracy for efficiency is especially useful in machine learning applications which allows a margin of error for various tasks. On top of that, many machine learning applications that take a massive amount of energy to train and execute can benefit the energy efficiency aspect of approximate computing.

In this paper, I will be studying the effect of approximate computing in an edge detecting algorithm called Sobel filter. Sobel filter has 18 multipliers and 17 adders. In [2] and [3], the authors present the energy profile of approximate adders and multiplies, meaning, the average relative error and energy efficiency of each approximate units. Therefore, the problem is to solve a classic combinatorial optimization problem like the one studied in [4] and [5]. In both these papers, the author presents Reinforcement Learning to solve such problems. I will be focusing specifically on using DQN and how it can be used in the aforementioned application of approximate computing. As far as I know, this is a novel approach in such an application. The reason I am inclining towards DQN instead of traditional Q-learning is that the number of states and actions pair in the above-mentioned problem domain is huge and therefore, it is very hard to keep and maintain a Q-Table. Therefore, I will also be studying paper on Q-learning such as [6].

## III. Project Timeline




## References


[1] V. Akhlaghi, S. Gao, and R. K. Gupta, "Lemax: learning-based energy consumption minimization in approximate computing with quality guarantee," in *Proceedings of the 55th Annual Design Automation Conference*, pp. 1–6, 2018.

[2] O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "Rap-cla: A reconfigurable approximate carry look-ahead adder," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 8, pp. 1089–1093, 2016.

[3] S. Vahdat, M. Kamal, A. Afzali-Kusha, and M. Pedram, "Letam: A low energy truncation-based approximate multiplier," *Computers & Electrical Engineering*, vol. 63, pp. 1–17, 2017.

[4] I. Bello, H. Pham, Q. V. Le, M. Norouzi, and S. Bengio, "Neural combinatorial optimization with reinforcement learning," *arXiv preprint arXiv:1611.09940*, 2016.

[5] T. D. Barrett, W. R. Clements, J. N. Foerster, and A. Lvovsky, "Exploratory combinatorial optimization with reinforcement learning," *arXiv preprint arXiv:1909.04063*, 2019.

[6] B. O'Donoghue, R. Munos, K. Kavukcuoglu, and V. Mnih, "Combining policy gradient and q-learning," *arXiv preprint arXiv:1611.01626*, 2016.