Policy Learning with Spiking Neural Network for Robot Manipulation Tasks

Osama M. Abdelaal & Florian Röhrbein*
Professorship of Neurorobotics, Chemnitz University of Technology, Chemnitz, Germany
*corresponding author: florian.roehrbein@informatik.tu-chemnitz.de

Abstract— This paper shows the results of deploying architecture for supervised biologically plausible neural networks based on spiking neuron models on ManiSkill2 Challenge. The aim is to corollate between the traditional artificial neural network and spiking neural network to train such an algorithm on advanced robotic tasks. The task chosen to be the testing environment represent by using a mobile arm robot to open a cabinet drawer, which it considers in the context of Human-Robot helping tasks as a human-like behavior. A comparison is established on the training and testing results and showed that, with applying 50k training steps, the success rate for the architecture that contains spiking neural network is 100% instead of 84% without it. To elaborate that through applying a customized supervised spiking neural network the training takes less time, less energy and testing results are more accurate.

Keywords— deep reinforcement learning, spiking neural network, robot manipulation tasks, behavior cloning, SNN, BC

Introduction

The advanced robotic task, such as handling different objects in different environments or controlling the movement of particular robot parts, requires integration among the sensory data that is received from different sensors. The new approaches that are used to achieve the advanced robotics' tasks relies totally on the neural networks. Although, there are two main drawbacks for using the Artificial Neural Network (ANN). First, the volume of sensory data is huge and constantly modified (not static), and it is varied based on the testing objective, environments, and the required action. Hence, it is not feasible to store this amount of data for training the algorithms, besides, the traditional algorithms of ANN cannot handle the high variation of data during the learning or training phase [1]. Second, the traditional ANN algorithms are developed based on biological neurons models such as McCulloch-Pitts Neurons model or the Perceptron model, this model interpreted by single activation value, which it is continues and static.

Based on the mentioned data parameters and the limitation of traditional ANN, the need for a different set of algorithms that is capable to overcome the challenges that confining the advanced robotics' tasks are upraised. The biological inspired algorithms that mimic the brain process characterized by using different sets of parameters, such as discrete spikes, spikes timing and spike rate, to transmit the sensory data are more comparable to solve the mentioned robotics' challenges. One of the most well studied and popular plausible neuron models is so-called Spiking Neural Network (SNN). The SNN currently considered as the most accepted theory to explain the information process through the brain. Moreover, SNN has a high ability to be implemented on a super-fast hardware (neuromorphic hardware) with the minimum energy consumption [2-4].

In this paper, an advanced robotic task achieved through SNN as a part of the training policy. The training pipeline included a stable supervised SNN as a replacement for ANN.

The Experiment

The experiment explained in this paper based on the environments and tasks (challenges) that is designed by The ManiSkill2 Challenge [5]. ManiSkill2 is a large-scale robotic manipulation benchmark, The challenge consists of object manipulation tasks, the tasks should be performed by a mobile robot arm in the SAPIEN simulation environment. It features more than 2000 diverse objects included in different 20 task categories (Fig.1), and a large-scale demonstration set in SAPIEN environment with fully physical and realistic simulator. Tasks are available in the challenge are: OpenCabinetDrawer, OpenCabinetDoor, PushChair and MoveBucket. For each task, there is an initial state and final state, the mobile robot arm must interact with the given object of a specific class (e.g. a cabinet) to change its state from initial (e.g. the upper drawer is closed) to achieve the final state (e.g. the upper drawer is opened). In this paper, the challenge OpenCabinetDrawer is only studied, and evaluated.

Fig. 1. ManiSkill environment, mobile robot arm and different challenge [5].

The architecture

As per the previous study [5]. The success rate is 76% on a single environment, and 37% when the original architecture trained on all environments. This suggests limitations, as per noticed, in the learning capacity of the architecture. The main architecture configuration to train the robot on mentioned challenge based on the point cloud pretrained dataset included 50k training step. The original configurations are kept as it is for a comparison purpose, and the network is changed.

The original network included the policy network that allows to implement several learning-from-demonstration algorithms such as BC (Behavior Cloning), BCQ (Batch-Constrained Q-Learning), CQL (Conservative Q-Learning), and TD3+BC. Besides, another online model-free agent's algorithms are provided such as SAC and TD3. The main training policy is identified as BC before the training process. BC applies supervised learning over state-action pairs provided by the demonstrator . The objective of BC is to find the parameter φ for which π_φ best matches the set of provided state-action pairs from the demonstrator. We find the parameter using maximum-likelihood estimation, i.e., we seek φ_* as (1) [7]. Besides the training policy, there are specific architectures built based on the network configuration during the policy and value network building processes.

$$\varphi^* = \arg\max\varphi \prod^{N}_{i=0} \pi\varphi(\tilde{}a_i | s_i) \qquad (1)$$

One of the main blocks of the architecture and it is built during the policy is point cloud-based architectures, which built of 1D & 2D convolutional layers connected with multi-layer perceptron (MLP) layers. This architecture is replaced with the proposed 2D convolutional SNN architecture, which consist of spiking dens layer working as pre-processing layer, connected to 3 convolutional SNN layers, consist of leaky integrate-and-fire (LIF) neuron. LIF formulate the neurons in a discrete part, hence a convolutional parameter is multiplied by the whole equation to convert it to discreet. The final layer is a fully connected layer represent the readout layer (Fig. 2), the main concept of this SNN architecture is adopted from [6].

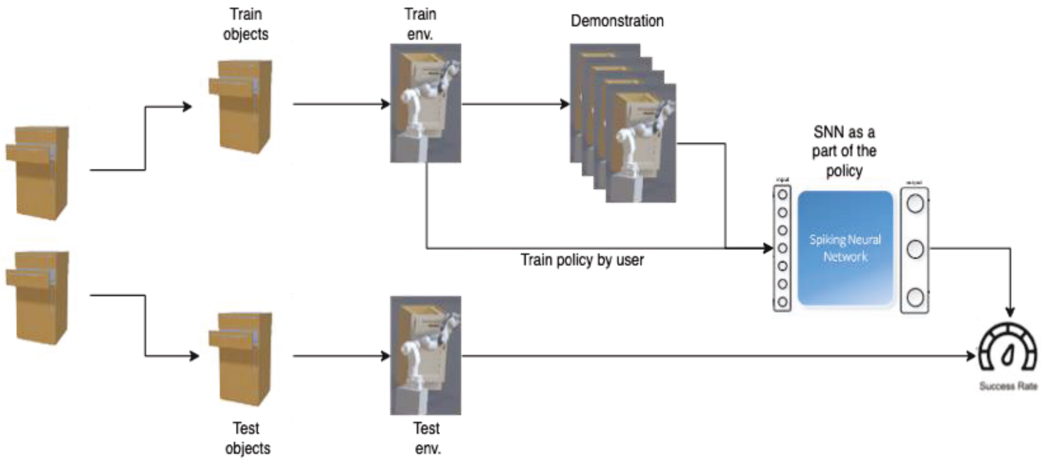

Fig. 2. The training/testing main architecture demonstrate the location of SNN

The results

The training for both models, the original model [5] and the customed model with SNN, are done on single NVIDIA RTX 3070 with 8 Gb ram which it takes 5 hours to complete the 50k training step (OneCycle 50k) with the original architecture and configuration [5], the policy network used is BC, the results showed 22% as policy absolute error (PAE) and a policy loss value (PL) is 0.0007, the model evaluation showed that the success or early stop rate is 0.5, which means that 84% the robot will success to open the upper drawer, or it fails before to achieve it. Besides it takes 10 seconds to achieve the task, Table 1.

By applied the SNN, with keeping all other parameters as same as the original, it is noticed that the experiment required 95 minuets to complete the training, which it is 32% of the original training time. Besides, the time to achieve the task (TTAT) during testing (Fig. 3) the robot arm could done the task in 2 seconds, with 6% as a policy absolute error and the success rate is 1.0.

TABLE 1. TABLE TYPE STYLES.

BC network policy	Training steps	PAE	PL	Training time(minuets)	TTAT (seconds)	Testing success rate
Cloud-based architecture	50k	22%	0.0007	300	10	84%
SNN architecture	50k	6%	0.00001	95	2	100%

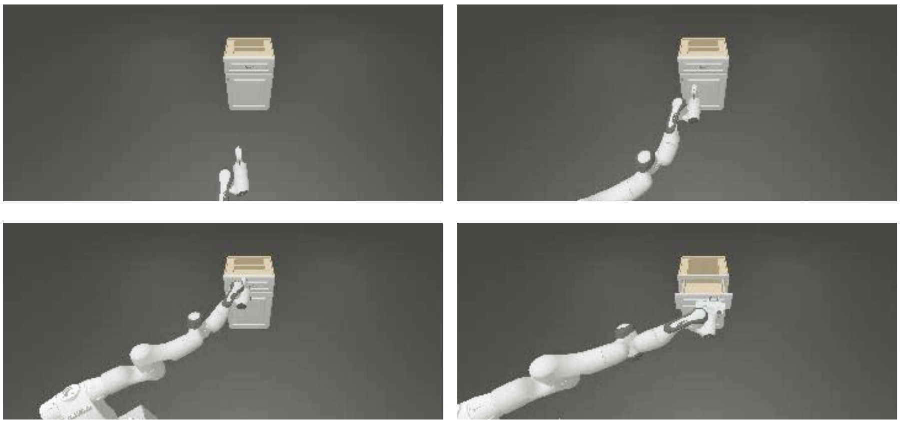

Fig. 3. The testing steps shows the process of achieving the task by the trained agent (the mobile robot arm).

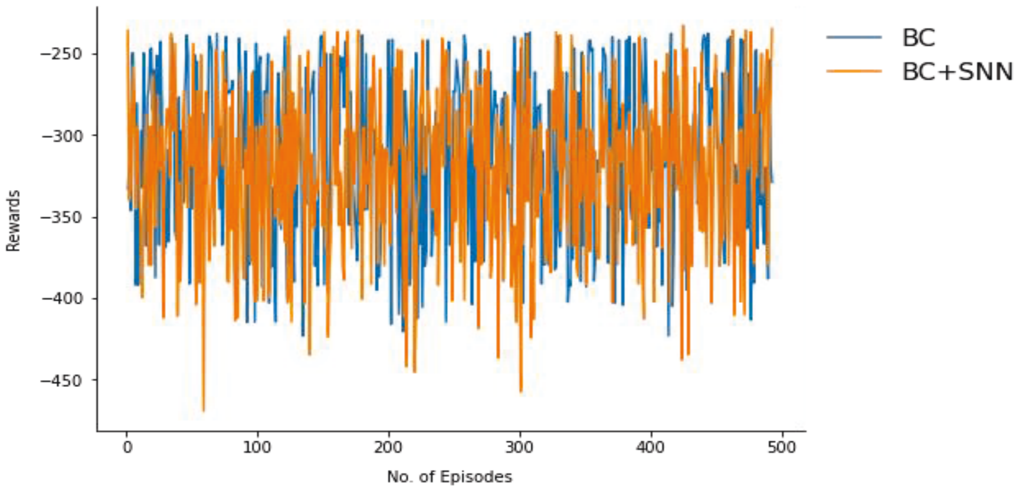

Fig. 4. The rewards values after evaluation.

The custom model (BC+SNN) is validated against the original BC model [5]. As it is noticed in Fig. 4, the validation results showed the rewards return values for the BC+SNN model is mostly higher than the values resulted from training the agent in BC model only, at the same episode. Which indicate that by applying the customized supervised spiking neural network, allows to improve the rewards values without any additional computational cost or time. That led to the higher testing success rate that resulted as a corresponding's experience for the agent after the training is completed. Nevertheless, it consumes less energy and produce better results.

CONCLUSION

In this experiment, a customed supervised spiking neural network designed and implemented as a part of training policy to teach an agent (the mobile robot arm) how to achieve a certain task (OpenCabinetDrawer) through BC (Behavior Cloning) algorithm in the framework of deep reinforcement learning. The ManiSkill environment and dataset are used to train the agent and evaluate the behavior of the trained agent. The validation results against the original BC model showed that using the SNN improved the success rate of the agent during the evaluation and used less time to train the agent comparing by the original architecture at the same numbers of time steps

REFERENCES

1. Kasabov, N.K. (2019). Time-space, spiking neural networks and brain-inspired artificial intelligence. Berlin, Heidelberg: Springer Berlin Heidelberg.
2. Belatreche, A., Maguire, L.P., McGinnity, M. (2007) Advances in design and application of spiking neural networks. Soft Computing 11(3):239-48.
3. Dorogyy, Y., Kolisnichenko, V. (2016) Designing spiking neural networks. In2016 13th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET) 2016 Feb 23 (pp. 124-127). IEEE.
4. Petro, B., Kasabov, N., Kiss, R.M. (2019) Selection and optimization of temporal spike encoding methods for spiking neural networks. IEEE transactions on neural networks and learning systems 31(2):38-70.
5. Mu, T., Ling, Z., Xiang, F., Yang, D., Li, X., Tao, S., Huang, Z., Jia, Z., Su, H. (2021). Maniskill: Generalizable manipulation skill benchmark with large-scale demonstrations. arXiv preprint arXiv:2107.14483.
6. Zimmer, R., Pellegrini, T., Singh, S.F., Masquelier, T. (2019). Technical report: supervised training of convolutional spiking neural networks with PyTorch. arXiv preprint arXiv:1911.10124.
7. Torabi, F., Warnell, G., Stone, P. (2018). Behavioral cloning from observation. arXiv preprint arXiv:1805.01954.