Circulated Deep Reinforcement Learning utilizing TensorFlow

P.Nandhini
Department of Computer Science
B.Tech,M.E, KGiSL Institute of Technology

Abstract- *Profound Reinforcement Learning is the mix of Support Learning calculations with Deep neural system, which had late achievement in learning confounded obscure conditions. The prepared model is a Convolutional Neural System prepared utilizing Q-Learning Loss esteem. The operator takes in perception, for example crude pixel picture and reward from the condition for each progression as info. The profound Q-learning calculation gives out the ideal activity for each perception and compensates pair. The hyper parameters of Deep Q-Network remain unaltered for any condition. Tensorflow, an open source AI and numerical calculation library is utilized to execute the profound Q-Learning calculation on GPU. The dispersed Tensorflow engineering is utilized to expand the equipment asset usage and lessen the preparation time. The utilization of Graphics Processing Unit (GPU) in the disseminated condition quickened the preparation of profound Q-arrange. On executing the profound Q-learning calculation for some conditions from OpenAI Gym, the operator outflanks a not too bad human reference player with few days of preparing.*

Keywords- Deep Reinforcement Learning, Tensorflow, Deep Q-Networks, Deep Q-Learning, Artificial Generalized Intelligence

I. INTRODUCTION

Reinforcement Learning (RL) is a branch of machine learning which takes its aspiration from behaviour psychology relating how software agents take action in an environment in order to maximize the cumulative reward. The formulation of the environment is done as Markov Decision Process (MDP) [1] where many reinforcement learning algorithms are based on dynamic programming techniques. Reinforcement learning algorithms can be trained to follow a dynamic programming path without actually having a perfect dynamic programming tree. RL is a special form of supervised learning where no correct input/output pairs can be generated, nor explicitly sub-optimal actions are presented. Further, all cases of RL are online while finding a perfect balance between exploration (of unknown territory) and exploitation (of existing knowledge). With recent exciting achievements in the areas such as deep learning, big data, increase in computational power (GPU) and new algorithmic techniques, the combination of reinforcement learning and Deep Neural Networks (DNN) has been successful.

II. RELATED WORK

The support learning specialist gets compensate comparing to each move made and performing just those activities that augment aggregate reward. The way toward learning can at that point be changed over into a managed learning issue where each perception is given as information, activity remunerate for each activity is given as yield and activity with greatest activity esteem is given to condition as following stage activity. The system, when prepared with a misfortune work, prepares the system to suggest an activity with a high level of exactness. The misfortune work is determined utilizing the Q-learning calculation which is utilized to prepare a particular sort of system called Deep Q-Network (DQN) [2]. The preparation process utilizes RMSprop streamlining agent to limit the misfortune.

The system is introduced utilizing Xavier introduction [3]. Amid the preparation procedure, the system is made to take up arbitrary activities with a specific likelihood so it investigates all conceivable powerful programming state space. This is proceeded for an extensive number of activities and after that the specialist is designed to play out an extensive number of voracious activities. In the wake of learning a vast number of scenes, the specialist figures out how to perform keeping pace with human dimension control. A similar operator is utilized to get familiar with a vast number of conditions with no adjustment in the system arrangement prompting a summed up specialist for support learning.

III. THEORY AND SYSTEM ARCHITECTURE

Q-Learning

Q-Learning is an esteem based RL calculation. The state-activity esteem capacity of Q(s, a) speaks to the greatest

future limited reward for picking the activity an in state s, and keep on acting ideally for all moves made later on.

$$Q(st, at) = \max Rt+1 \quad (1)$$

One can consider Q(s, an) as "the most ideal activity a for each state s in nature, to accomplish the most ideal state-activity esteem". The arrangement is to figure the estimations of Q(s, an) and Q(s', an'), at that point the activity with the best Q-esteem is picked according to the approach (2).

$$\pi(s) = \text{argmax}_a Q(s, a) \quad (2)$$

Here π speaks to the approach work, the standard how a particular activity is picked at a given state. So as to get the Q-work, think of one as change <s, a, r, s'>. The Q-estimation of state s and activity an as far as Q-estimation of the next state s' is given in (3).

$$Q(s, a) = r + \gamma \max_{a'} Q(s', a') \quad (3)$$

Deep Q-Learning algorithm

The neural systems are remarkable at approximating great highlights for entangled information designs. To speak to the Qfunction with a profound neural system, which takes four diversion states and activity as info and yields the relating Qvalue. This methodology has preference that if Q-values refresh are performed or the activity with most elevated Q-esteem is picked, at that point it requires just a single total forward go through the system and all Q-values for every one of the activities are accessible. Contribution to the neural system is the present perception (for example crude pixels) from the earth. In yield layer, it gives out the Q esteems for diverse activities. In this manner, number of yield units is equivalent to the number of activities in the earth.

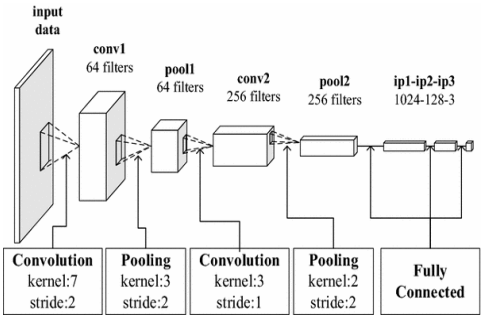

FIG:1:Convolution Neural Network Architecture.

This is a classical CNN with 3 convolutional layers, followed by 2 fully connected (FC) layers. The network input is four 84×84 grayscale game states. The network output is Q-values for each possible action. Q-values are real values, updated such that it can be optimized with squared error loss as given in (4).

$$L = \tfrac{1}{2} [r + \max_{a'} Q(s', a') - Q(s, a)]^2 \quad (4)$$

For all transition <s, a, r, s'>, the Q-table update rule in the

Q-Learning algorithm must be replaced as follows:

• Perform feed-forward pass for the state s in order to get

the predicted Q-values for all actions.

• Perform feed-forward pass for next state s' and calculate

maximum overall network outputs maxa' Q(s', a')

• Set the Q-value target for action a' to r + γ maxa' Q(s', a').

For all the other actions, set Q-value target to same as originally returned from 1st step, making the error 0 for those outputs.

• Update the network using the back-propagation.

System Architecture

The earth gives perception and reward at each progression to the operator. The perception is handled utilizing DQN which returns suitable activity esteem. The most ideal activity is picked and go back to nature. This procedure proceeds until nature ends. The loads of DQN are put away in a parameter server with the goal that a typical duplicate is held crosswise over imitated preparing frameworks. The misfortune is figured utilizing profound Q-taking in calculation from which the loads of DQN are refreshed.

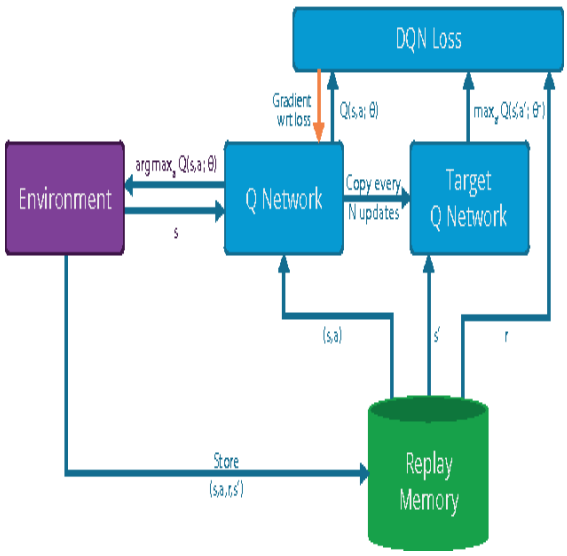

FIG:2:SystemArchitecture

IV. IMPLEMENTATION AND EXPERIMENTAL RESULTS

A.Tensorflow

TensorFlow For numerical calculation purposes, TensorFlow [11], an abnormal state library composed by Google Inc. is utilized. TensorFlow utilizes graphical model to play out the calculations which are less demanding and effective by and by. It is supported by Google Inc. what's more, an extraordinary network of open source givers. Multi-GPU support is accessible for preparing purposes and accumulation of AI and profound adapting abnormal state APIs for quicker advancement. Capacity to reestablish the program from checkpoints also, bolsters superior and out of the case conveyance of calculations in GPUs. The capacity to imagine the calculation diagram itself utilizing TensorBoard a web interface which makes a difference imagine scalars, histograms, pictures, sound and dispersions of tensors.

B. OpenAI-rec center

OpenAI rec center [12] is a broad toolbox for creating and looking at fortification learning calculations. All these conditions uncover a typical interface making it less demanding to attempt out different conditions against calculations.

C. Equipment and Software assets

The usage is done on a bunch with every hub setup (Intel Xeon 2 processors - 24 centers, 32 GB RAM, 500 GB HDD Storage, NVIDIA Tesla K20 GPU, 1000Mbps LAN). The product utilized is Ubuntu incorporates Python - for the calculations, HTML, CSS and JavaScript - for result perception, Low-level Libraries which incorporates CUDA Toolbox, NVIDIA cuDNN. Logical Python bundles incorporates TensorFlow, OpenAI-rec center, Flask and Numpy.

D. Assessment Metrics

• Average Loss per Episode: Loss is determined utilizing the Profound Q-Learning Algorithm. The normal misfortune per scene is utilized by RMSProp mentor to prepare the DQN operator and refresh the loads. It slowly diminishes over time as operator execution improves.

• Average Max Q-Value per Episode: Q-esteem measures how well the activity esteem for each activity are refreshed after some time. On powerful preparing, Q-esteem step by step increments.

• Duration per Episode: It indicates the quantity of steps the operator effectively plays in a scene. A decent operator can play the same number of scenes the earth gives. With viable preparing the operator makes more reward per scene, the term per scene step by step increments.

• Total Reward per Episode: The operator takes ideal activity for current state and ensures that compensate is augmented. The reward is the earth's measure on specialist's execution to boost number of ideal steps. With successful preparing, complete reward per scene progressively increments.

E. Exploratory Results

The execution of DQN calculation is tried on an aggregate of 8 situations given by OpenAI-rec center.the current state of the model for determining how to update the parameters and doesn't consider previous parameter values. The process for categorizing input images, comparing predictions with the correct categories, calculating loss, and adjusting parameter values is repeated many, many times. Computing duration and cost would quickly escalate with larger and more complex models, but our simple model here doesn't require much patience or high-performance equipment to see meaningful results.

Environment	Average MaxQ Value	Time Duration	Total reward
Assault-v0	5.734	10100	81
BeamRider-v0	2.9	7000	100
MsPacman-v0	19.91	1438	116
Pong-v0	0.56	6299	9.5
Qbert-v0	11.9	726	52
SpaceInvadersv0	6.167	1737	50

Table:1:Result Analysis

V. CONCLUSION

The execution of profound Q-learning calculation worked effectively with little conditions with predetermined number of dynamic programming states. The outcome and

investigation from the venture have appeared profound Q-learning calculation can be utilized to sum up the working of independent frameworks in the genuine world by learning the earth. Out of eight conditions, the operator has effectively taken in the state space for seven situations. The execution of the specialist totally relied upon the preparation procedure and the system hyperparameters. The utilization of GPU quickened the usage and working by ten times. The undertaking usage has numerous conveyed specialists and along these lines can be prepared at the same time. This is helpful as the specialist joins to the best activity esteem in exceptionally brief time.

REFERENCES

[1] R. S. Sutton and A. G. Barto, Reinforcement Learning : An Introduction, 2nd Edition ed., M. Press, Ed., Massachusetts Ave, 2016, pp. 47-74.

[2] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra and M. A. Riedmiller, "Playing Atari with Deep Reinforcement Learning," CoRR, vol. abs/1312.5602, 2013.

[3] D. Silver and J. Heinrich, "Deep Reinforcement Learning from Self-Play in Imperfect-Information Games," CoRR, Vols. abs/1603.01121, 2016, 2016.

[4] V. Mnih, K. Kavukcuoglu and D. Silver, "Human-level control through deep reinforcement learning," Nature, vol. 518, pp. 529-533, February 2015.

[5] M. G. Bellemare, Y. Naddaf, J. Veness and M. Bowling, "The Arcade Learning Environment: An Evaluation Platform for General Agents," CoRR, vol. abs/1207.4708, 2012.

[6] H. van Hasselt, A. Guez and D. Silver, "Deep Reinforcement Learning with Double Q-learning," CoRR, vol. abs/1509.06461, 2015.

[7] B. Bakker, "Reinforcement Learning with Long Short-Term Memory," in In NIPS, MIT Press, 2002, pp. 1475-1482.

[8] T. Shankar, S. K. Dwivedy and P. Guha, "Reinforcement Learning via Recurrent Convolutional Neural Networks," CoRR, vol. abs/1701.02392, 2017.

[9] J. Dean and G. S. Corrado, "Large Scale Distributed Deep Networks," Proceedings of the 25th International Conference on Neural Information Processing Systems, pp. 1223-1231, 2012.

[10] C. Fernando, D. Banarse, C. Blundell, Y. Zwols, D. Ha, A. A. Rusu, A. Pritzel and D. Wierstra, "PathNet: Evolution Channels Gradient Descent in Super Neural Networks," CoRR, vol. abs/1701.08734, 2017.

[11] D. Jeffrey, "TensorFlow: A System for Large-Scale Machine Learning," Google Brain, 2015. [Online]. Available: www.tensorflow.org. [Accessed 9 April 2017].

[12] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang and W. Zaremba, "OpenAI Gym," OpenAI, 5 June 2016. [Online]. Available: gym.openai.com. [Accessed 9 April 2017].

[13] P Ajay Rao , Navaneesh Kumar B , Siddharth Cadabam, PraveenaT,"Distributed Deep Reinforcement Learning using TensorFlow", International Conference on Current Trends in Computer,Electrical, Electronics and Communication[Accessed2017