Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 13, Issue 1, January 2022: 946-954

Optimizing Deep Reinforcement Learning with a Hybrid Multi-Task Learning Approach

M. Lubna Yasmeen[1], Qurratul Aini[2], Sumaya Tazeen[3], Sumaiya Anjum[4]

[1]Assistant Professor, Sreyas Institue of Enginering and Technology, Hyderabad
[2]Research Scholar, Shri Jagdishprasad Jhabarmal Tibrewala University, Rajasthan
[3]Assistant professor, Shadan Women's College of Engineering and Technology, Hyderabad
[4]Assistant Professor, Shadan Women's College of Engineering and Technology, Hyderabad

Abstract: Deep Learning is now a popular representation learning strategy for many forms of ML, including reinforcement learning, because to improvements in AI (RL). Thus, deep reinforcement learning (DRL) was born, which blends deep learning's enormous ability for representational learning with more traditional reinforcement learning techniques. This novel technique has unquestionably played an important role in improving the performance of model-free intelligent RL systems. This method of improving performance was only tested on intelligent systems utilising reinforcement learning algorithms that can only learn one task at a time. A lack of data efficiency was proven when such intelligent systems were expected to work in overly complex and rich data environments, and this was especially true when single-task learning was used. As a result of present technology limitations and the related demands on several operating systems, this was inevitable. This might be solved by using multi-task learning. When optimising deep reinforcement learning agents in two different but semantically equivalent environments with related tasks, we use a technique called parallel multi-task learning (PMTL). There should be a worldwide network of actor-critic models that may exchange their cumulative expertise in various situations in order to enhance performance.

Keywords— Transfer learning, Multi-tasking, Actor-mimic, Deep reinforcement learning

I. INTRODUCTION

Reinforcement learning has been a key focus in robotics and intelligent agents during the last several decades. Many elements must be considered while constructing real world robots in order for them to maximise their reward at every time step t [2]. RL has emerged as a significant ML prototype that controls how an RL agent should act best in its operational environment due to recent advances in the area of machine learning. RL-based systems exhibit weak performance metrics when compared to other ML classes, such as supervised learning and unsupervised learning. As a consequence, the DRL agent was unable to come up with a suitable strategy for its surroundings. In the wake of the discovery of Deep Learning (DL) and its representational learning capabilities, known as Deep Reinforcement Learning (DRL), a new route in reinforcement learning has been opened up (DRL). Continuous action control, 3D first-person settings, and gaming have been more popular with the introduction of DRL-based systems. If you've ever played Atari or chess or go, you'll know that DRL-based models can beat you at these games. No matter how well they are at a single task

learning system, intelligent agents perform only fair or average in more data-rich and sophisticated games like three-dimensional shooters, contrary to common perception.

Multi-task-based learning may improve the RL agent's performance in this situation. Numerous intelligent agents would simultaneously investigate and analyse a collection of actions closely linked to the operational environment. A DRL algorithm, such as A3C, will govern the activities of each of these agents (Asynchronous Advantage Actor-Critic). As a result, the global network receives the network parameters of each neural network agent on a regular and asynchronous basis. Global network parameters are recalculated when all learning parameters are acquired by all agents in the network. A limited number of agents will be given access to the data in the future. It is the primary goal of this stage to increase agent performance in a unique context via the use of positive information and shared learning opportunities. Parallel-based multi-task learning [5] is widely accepted as the most efficient approach for learning many tasks at once. [7,8]. A single learner (also known as a critic) is coupled with a large number of actors in the DRL paradigm. For students (or critics), network parameters allow them to exchange data with actors who build their own learning courses or traversal routes. As a result of this, actors will be given the most up-to-date parameter list before the next learning cycle begins. Each agent's previous acts are transmitted to one another using the aforementioned approach. As a result, the RL intelligent agent's learning rate would be accelerated.

The remainder of the paper is broken out as follows: Distral, iMPALA, and PopArt were investigated in great detail in the second portion. We go into further depth in Section III on the hybrid multi-task learning paradigm that we've developed. It is shown in Section IV that the hybrid multi-task learning technique is effective. Section V comprises all experiments and their outcomes, as well as specifics and discoveries. Conclusions and future research proposals are provided in Section VI.

II. RELATED WORK

Developed by DeepMind, Distral is a multi-tasking training system RL prototypes that can handle a large number of simultaneous tasks. It was critical to this model's development that it be able to extract and then transform the behaviour patterns of several people working in a multi-tasking real-life situation. The major emphasis of Distral's design method is on a distinct policy for an individual employee; this policy takes into account the similarity in behaviour across related professions rather than creating a strategy for parameter sharing. Regularization and Kullback-Leibler (KL) divergence may be used to derive the distilled policy [7]. One project's findings may be summarised into a single policy that could be applied to other experiments in the same context.

If the organisation employed this technique, which enables workers to teach one another how to address their own problems, they may be more committed to the company's overall policy. A distillation method will be used to teach this policy, which acts as a model for all individual task policies. Researchers have found this method particularly effective for transferring information across challenging 3D operational environments associated with RL challenges.

Experimentally, the Distral approach beats more conventional methods that depend on sharing neural network parameters for multi-tasking and transfer learning. Listed above are some of the primary causes behind this. Because distillation has such a profound effect on optimization, this is the

primary reason. A distilled model is more common when KL divergences are used as the primary way of regularising task models' output. To train specific task models in an environment utilising regularisation, the distilled model may be employed. The regularisation of personnel collection in the distilled model is based on the idea that task limits may have a significantly greater effect than parameters alone when enforced [8]. [9, 10].

Another well-known multitask learning technique is IMPALA (Impact Weighted Actor-Learner Architecture), a Google DeepMind algorithm. One set of parameters is thought to be sufficient to build an RL agent in a distributed agent architecture. The IMPALA model's design is based on the fact that it may be implemented in any workplace. As well as being efficient on a single system, this approach may be expanded to include several machines without sacrificing data economy or resource utilisation.

Impala is able to maintain a steady trajectory of learning with high throughput because to the unique off-policy correction mechanism V-trace, which incorporates both decoupled action and learning Using the DRL paradigm, a small group of performers is joined by a single learner (sometimes referred to as the critic). Each actor in the ecosystem receives the learner's (critic's) learning trajectory through a queueing mechanism. Since it comes from so many diverse sources, a student who has learned this knowledge may use it to develop a foundational policy. [9] Actors (critic modules) get new policy parameters from the learner in the next learning cycle (trajectory).

There are several similarities between RL's A3C algorithm and this one.

The IMPALA's design was greatly influenced by the algorithm. Agents and learners in IMPALA work together to develop knowledge in the system's architecture. The IMPALA's design generates n policies and Vn as a baseline based on an actor-critic paradigm. This system's basic components are a group of people that constantly produce new paths of exploration. One may learn about n.'s off-policy policy by following the path of numerous players. One or more pupils may exhibit this kind of behaviour. New learning policies are introduced at the beginning of each trajectory.

After n phases, each actor would put this strategy into action within the confines of their own operational situation. The learner receives a new collection of data, including a trajectory of states, actions, and rewards, as well as distributions of policy alternatives, once each actor has completed these stages. As long as the environment's members provide their trajectory data, the learner will be able to make regular policy changes. There are several ways in which people living in an ecosystem might share their information and experiences with a central module for evaluation. An autonomous agent and additional learners are included in the model when the core learner calculates gradients. Operational flexibility enables actors to be present on the same machine or spread out among multiple machines, depending on their needs.

Google DeepMind rectified the existing IMPALA model's weaknesses with a third technique named PopArt. PopArt was designed to improve RL in multi-tasking contexts by addressing the root causes of poor performance. Distraction reduction and learning process stability are two of PopArt's primary goals for the IMPALa model. This will make it easier to use multi-task RL approaches in the future. Just a small fraction of activities that learning algorithms have to complete are thought to be a source of "distraction" for them. As a result, competition for scarce resources is fierce. Do you think it is

possible to strike a balance between the conflicting demands of various educational pursuits? Long-short term memory (LSTM) recurrent neural networks are used to combine many CNN layers with additional approaches like word embeddings in the Pop Art model development process.

The RL agent and each task's trajectory are pooled in order to make the PopArt model work. It is because of the pop art model that each agent has a distinct job and an equal influence on the overall dynamics of learning. In the PopArt model's design, the neural network's weights may be shifted depending on the outcomes of all the actions in the environment. In PopArt, ultimate results, like a game's score, are calculated using the mean and standard deviation. Using these predicted values, PopArt will next normalise its goals before completing the process by updating the network's weights. As a result, learning has become stronger and more long-lasting. PopArt's capabilities and advantages over earlier multi-task RL systems were shown via trials including popular Atari game settings.

III. PROPOSED HYBRID MULTI-TASKING LEARNING METHOD

Some of the primary challenges associated to DRL multitasking have not yet been completely addressed by the state of the art, hence the proposed hybrid multitask learning technique was established. Due to these issues and limitations, we have devised an approach to reducing both the quantity of training data samples required to get acceptable results and the time required for exploration [12].

Two operating systems with a high degree of semantic similarity have been deployed in an attempt to address the issues outlined in the preceding paragraphs. One of the most essential features of the A3C technique is its ability to concurrently learn several instances of a single target work and then use that knowledge to improve the model's performance. Two distinct but semantically equivalent contexts and accompanying activities were used to develop a unique hybrid multi-task learning approach to achieve this aim. Here's how to go about it. The proposed hybrid approach is based on the A3C algorithm's ability to apply to semantically related actions happening in two distinct locations. A3C is utilised to keep track of all of the different instances of each task running in each environment when using this hybrid multi-task learning approach.

Threads A and B are subtasks that pass on specialised information to students from across the globe, and this is an international network of knowledge (A). As a result of this, the student will design and implement a new policy for the agent workers' threads. Multi-task learning and deep reinforcement learning are the ultimate goals of this model's hybrid multi-tasking. In Fig. 1, this hybrid multitasking technique is shown at a high level.

Multi-threaded asynchronous A3C algorithm modifications are used in the hybrid model of multi-tasking learning. In order to speed up the training of deep neural network rules, this model was created. At this time, we're using eight asynchronous actor-learners in conjunction with a Windows 10 cloud server machine equipped with graphics processing units (GPU). It is possible to adopt an asynchronous approach to policymaking when a large number of actors-learners are working on different aspects of the environment at the same time. There are a variety of guidelines that may be employed by individuals to maximise this variety.

When many actor-learners simultaneously update global network parameters asynchronously, the correlation between these changes is expected to be lower.

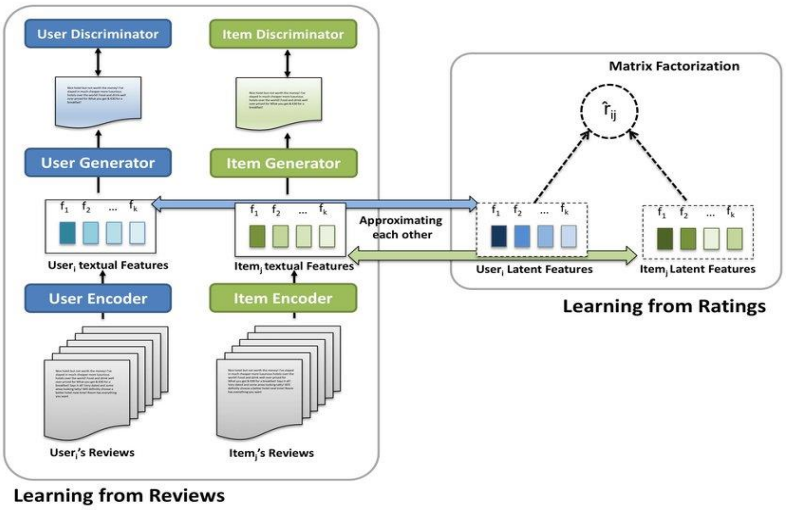

Figure1: Architecture of the Proposed Hybrid Multi-Task Learning Model

Using a single system with multi-core CPU and GPU capabilities, this strategy performed well on a range of Atari 2600 domain games. The most important component in achieving these goals is the semantic similarity of critical actions in the two distinct game environments. In the reinforcement learning discipline, multitasking is recognised to be difficult because of the negative transfer. To help another agent improve their performance on the goal job, they may learn from a collection of source samples provided by another agent. "Multi-task learning" is the term for this. The way information is delivered may have a good or negative impact on the learning process and the performance of the agent. This effect is more likely to occur when the source and target occupations are more different. It's possible that a person acquiring information will have a favourable or negative response to it.

Partial observation may be boosted by exchanging learning across operational settings that are semantically equivalent. In addition, multiple actor-critic models in two semantically related scenarios would help RL agents in terms of exploration, raining samples, and training time .

IV. PROTOTYPE IMPLEMENTATION

When implementing the proposed hybrid multi-task learning model for the rest of this project, a prototype of A3C is developed here. Various Open AI Gym prototypes were put to the test on an Atari 2600 during development. Open Al's Gym module may be used to design and compare RL algorithms. A new method to multitasking, Breakout-A3C v0, was developed from the ground up. The actor-critic technique is used for the model's high-level design. A neural network validation model might be used to approve and govern the Atari 2600 game environment. Using two CNN models at the root level, this environment will be able to construct actor and critic modules for one worker.

A huge number of CNN class instances are required to manage the enormous number of workers in the multitasking paradigm. Two CNNs have been set up to assist in the global distribution of actor-

critic modules on networks. The multitasking paradigm is supported by the global network represented in Figure 2. This list includes each and every one of them.

The CNN block is named for both the actor and the critic (value function) way a module is put together). Here's a simpler way to say it: More than one person attacking the same game area simultaneously regardless of how they're put together. There is a good chance this is the case. All of us begin at a different point in our lives. This means that everyone will perceive the universe from a distinct perspective.

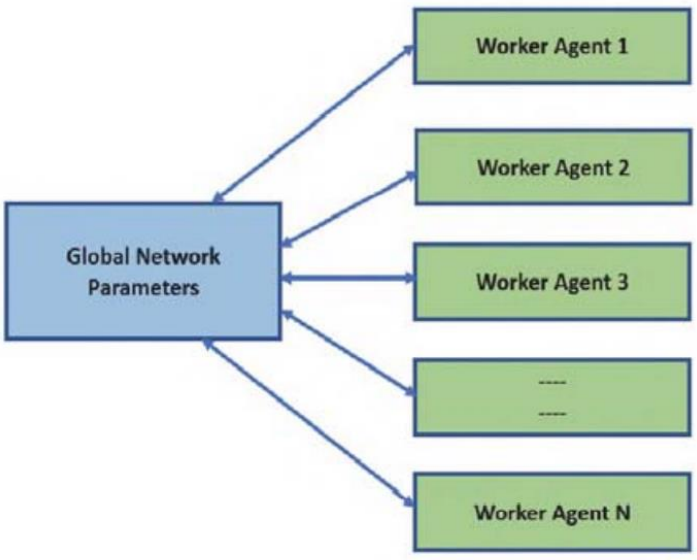

Figure 2. A3C Multi-Task Worker Agents Model

Each worker agent is required to play a certain number of episodes in order to determine its own policy loss. Losses in neural network operation affect the gradient values of these modules, both actors and critics. After the work agent has finished a certain number of game episodes, the gradient values will be made available to the whole network.

$$a = sample\ an\ action\ a \sim \pi_\theta(a|s)$$

$$s`, r, done = Perform\ action\ a - env.step(a)$$

$$G = r + \gamma V(s`)$$

$$L_p = -(G - V(s))\log\left(\pi(a|s, \theta_p)\right)$$

$$L_v = (G - V(s))\text{^}2$$

$$\theta_p = \theta_p - \alpha * d_{L_p} / d_{\theta_p}$$

$$\theta_v = \theta_v - \alpha * d_{L_v} / d_{\theta_v}$$

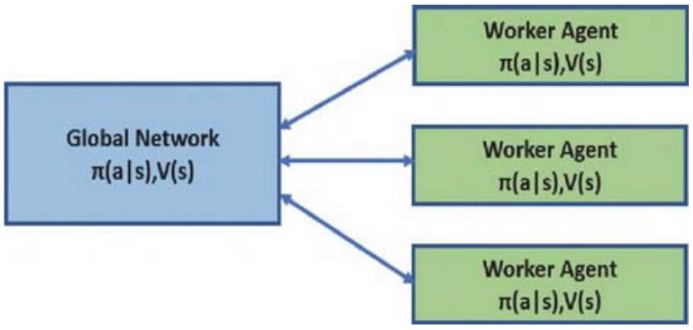

Figure 3. Parameter Sharing of Global Network with Worker Agents

In a global network of multitasking agents, the actor module and value network are both represented by two neural networks. Using a worldwide network of worker agents, each of whom has its own copy of the policy and value network, weights will be routinely distributed. After a few episodes, employees may use their network weights to compete against one another. An experienced worker agent may create their own policy gradient changes and value updates. The new information will be available to these employees. Global networks will be able to change the weights of their components depending on the gradients of the gradients of their worker agents. The global network's parameters are regularly updated and returned to it in order to maintain the worker agents working with the most up-to-date settings. By simulating different game settings in worker threads and then performing each step of the gameplay, the global network may then calculate and broadcast corrected gradients to each worker thread.

V. EXPERIMENTS AND EVALUATION RESULTS

Reinforcement learning may benefit from A3C's multi-threaded and asynchronous approach to computing. While memory replay is avoided via sparse data, this strategy allows models to be trained by doing numerous distinct investigations on the same task [11]. The Atari 2600 will be used to evaluate the hybrid multi-task learning paradigm. It was decided to test the suggested technique using a variety of different video games including BeamRider 4, Breakout 0, BeamRider 0, as well as Pong 0. The initial assessment phase will focus on assessing each of these gaming settings individually in order to obtain data. Images 6 to 9 in the gallery show the A3C test results for each ATARI 2600 game based on the multi-task worker model. Tools for displaying data the charts were created using Tensor Board from TensorFlow. It is possible to see the agent's progress on the x-axis and the prizes he or she has obtained on the y-axis (game score). There are many of the same ideas that apply to figures 4-7.

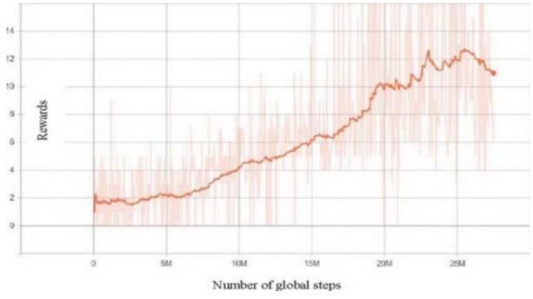

Fig. 4. Breakout-vO Test Results with 8 Workers

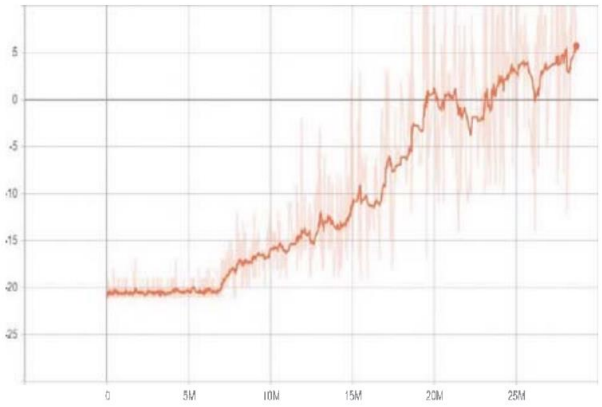

Fig. 5. Pong-v0 test Results with 8 Workers

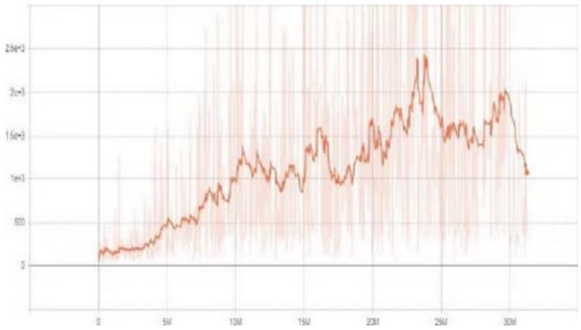

Fig. 6. DemonAttack-v0 Test Results with 8 Workers

Educators and gamers throughout the world may use a multi-task learning paradigm to integrate the knowledge gained from both gaming contexts at the same time. Take a look at Figure 10 for an illustration. By using one global network, a hybrid multitasks model's influence on performance in various contexts with high semantic similarity may be examined and verified.

VI. CONCLUSION AND FUTURE WORK

The hybrid multi-task learning method we used in our research is beneficial to DRL agents. This approach may include multi-task learning from DRL agents that operate in two different but semantically identical contexts on related tasks, as shown by our experiments. It's done this way. During the early stages of the investigation, the DRL algorithm A3C is used to analyse Atari 2600 game settings. Hybrid multi-task learning was shown to be capable of learning a range of gaming tasks at the same time while without sacrificing the performance of any one agent. The semantic closeness of the associated activities, even if they are carried out in separate places, minimises the transfer of negative information. An increasing number of worker threads will be added to the hybrid A3C model over time, putting it to the test. When doing multi-task reinforcement learning in a GPU cloud-based machine environment, we also want to minimise the negative effects of knowledge transfer and catastrophic forgetting.

REFERENCE

1. H. Robbins and S. Monro, “A stochastic approximation method,” The Annals of Mathematical Statistics, pp. 400–407, 1951.

2. P. Jain, S. Kakade, R. Kidambi, P. Netrapalli, and A. Sidford, “Parallelizing stochastic gradient descent for least squares regression: mini-batching, averaging, and model misspecification,” Journal of Machine Learning Research, vol. 18, 2018.
3. D. F. Shanno, “Conditioning of quasi-Newton methods for function minimization,” Mathematics of Computation, vol. 24, pp. 647–656, 1970.
4. J. Hu, B. Jiang, L. Lin, Z. Wen, and Y.-x. Yuan, “Structured quasinewton methods for optimization with orthogonality constraints,” SIAM Journal on Scientific Computing, vol. 41, pp. 2239–2269, 2019.
5. J. Pajarinen, H. L. Thai, R. Akrour, J. Peters, and G. Neumann, “Compatible natural gradient policy search,” Machine Learning, pp. 1–24, 2019.
6. J. E. Dennis, Jr, and J. J. Mor´e, “Quasi-Newton methods, motivation and theory,” SIAM Review, vol. 19, pp. 46–89, 1977.
7. J. Martens, “Deep learning via Hessian-free optimization,” in International Conference on Machine Learning, 2010, pp. 735–742.
8. F. Roosta-Khorasani and M. W. Mahoney, “Sub-sampled Newton methods II: local convergence rates,” arXiv preprint arXiv:1601.04738, 2016.
9. P. Xu, J. Yang, F. Roosta-Khorasani, C. R´e, and M. W. Mahoney, “Subsampled Newton methods with non-uniform sampling,” in Advances in Neural Information Processing Systems, 2016, pp. 3000–3008.
10. R. Bollapragada, R. H. Byrd, and J. Nocedal, “Exact and inexact subsampled newton methods for optimization,” IMA Journal of Numerical Analysis, vol. 1, pp. 1–34, 2018.
11. L. M. Rios and N. V. Sahinidis, “Derivative-free optimization: a review of algorithms and comparison of software implementations,” Journal of Global Optimization, vol. 56, pp. 1247–1293, 2013.
12. A. S. Berahas, R. H. Byrd, and J. Nocedal, “Derivative-free optimization of noisy functions via quasi-newton methods,” SIAM Journal on Optimization, vol. 29, pp. 965–993, 2019.