Robot moving using Trust region policy optimization compared to proximal policy optimization:

Trust Region Policy Optimization (TRPO) is an algorithm for optimizing control policies in reinforcement learning. It uses a trust region method to ensure that the update to the policy is conservative and does not make the policy worse. TRPO uses a second-order approximation of the policy's performance to determine the size of the trust region, which helps to ensure that the policy update is always improving the policy's performance.

Proximal Policy Optimization (PPO) is also an algorithm for optimizing control policies in reinforcement learning. It uses a technique called "proximal optimization" which involves optimizing a "surrogate" objective function that is similar to the true objective, but is easier to optimize. PPO uses a "clip objective" which helps to ensure that the policy update is always improving the policy's performance and also it helps to avoid large policy updates that can make the policy worse. PPO is considered more sample efficient than other algorithm like TRPO, Asynchronous Advantage Actor Critic (A3C), and others.

Trust region method:

This state of the art relay on the trust reign method and in the policy gradient families there are two famous algorithms uses trust region method to find optimization, proximal policy optimization (PPO) and trust region policy optimization (TRPO), this method help us to increase the sample efficiency and the reliability of finding the optimal policy.

How TRPO find the optimization:

Trust Region Policy Optimization (TRPO) is an algorithm that optimizes the policy by making small updates to it, with the help of second-order approximation called "Fisher information matrix" to determine the size of the update. This ensures that the updates are always improving the policy's performance without making it worse, and find a point inside the region with the higher optimization, The algorithm continues to update the policy until it reaches a locally optimal solution, with TRPO looks at how much a change in the policy would affect the performance, if this change is too big, it will reduce it and make the update, if it's too small, it will increase the update size until it reaches an optimal solution. See the fig.1.

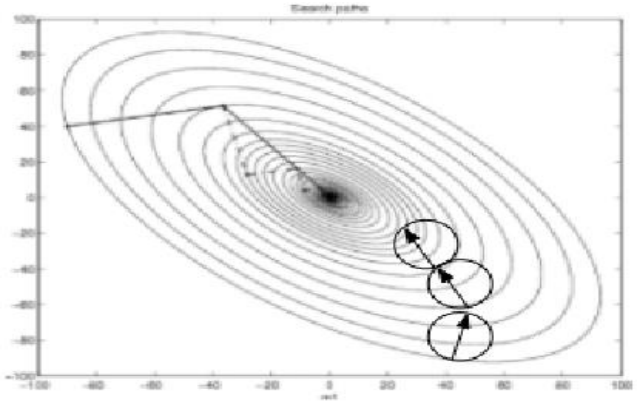


Fig.1. Trust Region Method

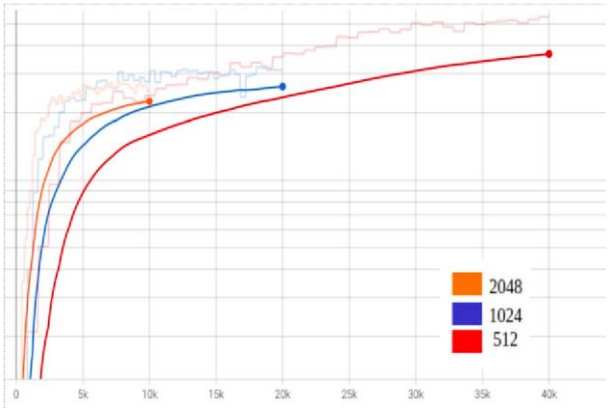
ANT robot experience:

We will use the Ant environment to train two agents, one using the Trust Region Policy Optimization (TRPO) method and the other using the Proximal Policy Optimization (PPO) method. The training will be done on the Google Colab platform using PyTorch toolkits. We will use Tensorboard to visualize the results and compare the performance of the two agents. Google Colab is a platform that allows us to execute any Python code and has the resources of a Linux operating system, a GPU processor, and 27.3 gigabytes of RAM.
We will compare the average return for each environment to the average return obtained using the Proximal Policy Optimization (PPO) method after training this robot environment, We will use the Tensorboard tool to visualize the results. In addition, we will also analyze how the optimization is impacted by the hyper-parameters during the TRPO training process.

As previously stated, the policy gradient approach aims to optimize the agent's performance. We can further optimize the performance by choosing the best hyper-parameters, such as the training rate, epsilon greedy, batch size, discount factor, and number of epochs. In this experiment, we will focus on the batch size and its impact on the efficiency of the Trust Region Policy Optimization (TRPO) method when using the Ant environment.

The processor CPU, GPU, TPU, and RAM have a significant impact on the batch size efficiency, as seen in Fig. 2. Smaller batch sizes result in more steps or iterations per episode because one episode requires visiting all the dataset's data, and one step requires one batch. The results of the different batch sizes for TRPO and PPO are that higher batch sizes (2048 and 1024) result in faster learning but later convergence and stability. The processor and RAM in this experience are fast, which leads to good response for a high batch size of 512.

Increasing the batch size hyper parameter can lead to increased efficiency in the training process. However, it may also reduce stability during the training, or the stability will be achieved after a longer number of training epochs. It's important to balance the efficiency and stability by finding the optimal batch size that fits your specific use case and resources.



Y-axis represents the average return and the x-axis represents the number of training epochs, the graph would show how the average return changes as the number of training epochs increases. It would give an understanding of how the agent's performance improves over time, and how the agent's performance is affected by different batch sizes. The graph can be used to determine the optimal number of training epochs, and the optimal batch size that gives the best balance of efficiency and stability.

Fig. 2. Results of the average return that the neural network model produced for the various batch sizes, TRPO method,

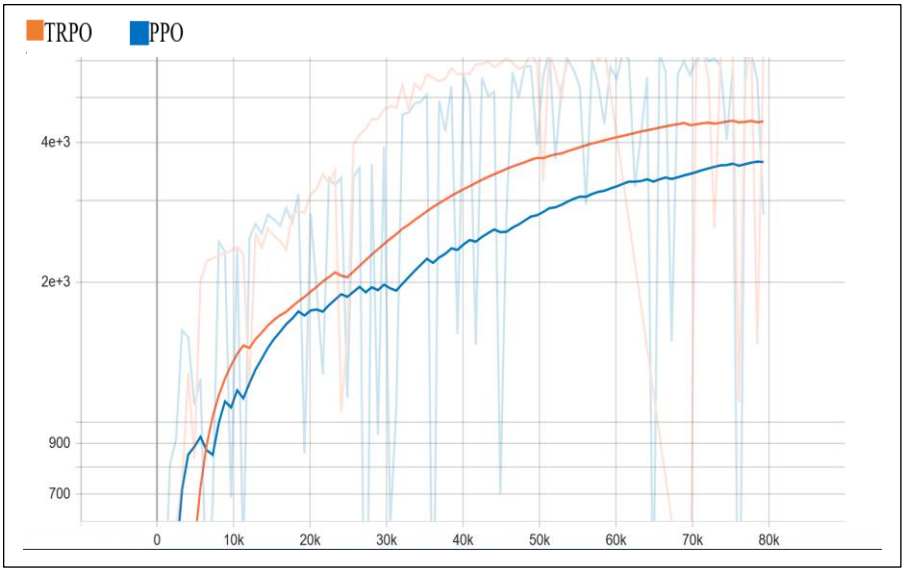
The training results:

TRPO and PPO are two algorithms that use two neural networks, an actor network and a critic network, to optimize the policy. The actor network chooses the action to take, while the critic network estimates the value of the action selected by the actor. The critic network calculates the TD error to improve the network parameters and then sends this error to the actor network to improve its parameters for a more accurate prediction of the next action. TRPO has the following parameters: actor learning rate = 0.001, critic learning rate = 0.0001, γ=0.99, λ =0.95, Dkl = 0.25. PPO has similar parameters except ε =0.3 and the entropy coefficient = 0.1. The number of episodes was 1000.

Table.1. Experience Average Return

Environment	TRPO Average Return	PPO Average Return	TRPO Training time	PPO Training time
Ant	4442	3630	1 Hrs., 52 Min.	58 Min.

In the fig. 3.The TRPO and PPO methods provided high performance when training an ant robot, resulting in the robot's ability to move forward easily and quickly. TRPO, considered currently the best in its class, performed better than other methods but took longer to produce results. Despite the longer time, the end results were of higher quality.



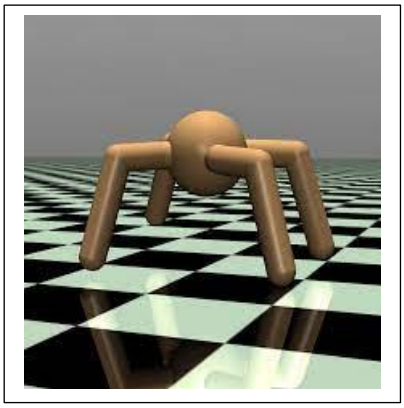


Fig.3. Ant Environment Result

References:

[1] R. S. Sutton and A. G. Barto, An Reinforcement Learning: Introduction. Mit Press, 2012.

[2] M. Sewak, Deep reinforcement learning: Frontiers of artificial intelligence, 1st ed. Singapore, Singapore: Springer, 2020.

[3] Ott Toomet, “Stochastic Gradient Ascent in maxLik,” 2020.

[4] Jorge Nocedal, Stephen J. Wright, Sequential Quadratic Programming, Springer, New York, NY, 1999.

[5] Iris Smit, Reinforcement Learning, and surrogate reward functions based on graph Laplacians, Utrecht University, 2022

[6] K. Lange, MM Optimization Algorithms. New York, NY: Society for Industrial & Applied Mathematics, 2016.

[7] C. Canuto and A. Tabacco, Mathematical analysis II, 2nd ed. Basel, Switzerland: Springer International Publishing, 2015

[8] “PyTorch Lightning,” Pytorchlightning.ai. [Online]. Available: https://www.pytorchlightning.ai/. [Accessed: 03-Oct-2022].

[9] “Google colab,” Google.com. [Online]. Available: https://research.google.com/colaboratory/faq.html. [Accessed: 03-Oct2022].

[10] Machinelearningmastery.com. [Online]. Available: https://machinelearningmastery.com/difference-between-a-batch-and-anepoch/. [Accessed: 03-Oct-2022].