Aeliya Grover

Dr. Nelson

ATCS: Neural Network, Period 4

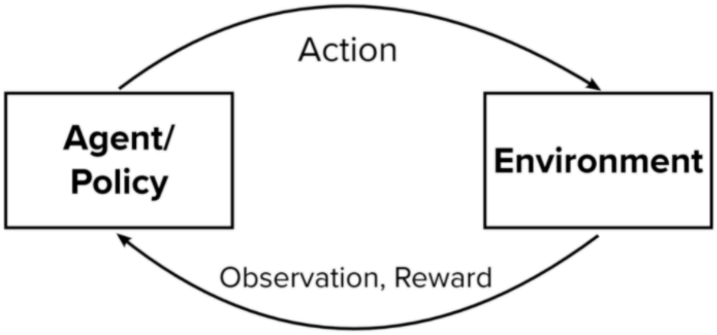
December 4 2023

An Introduction to Proximal Policy Optimization

Proximal Policy Optimization (PPO) is a Deep Reinforcement Learning (deep RL) algorithm developed by OpenAI in 2017 that uses rewards to transform behavior and improve the actions of a neural network (Schulman *et al*. 2017). In the field of Psychology, Operant Conditioning is a process where rewards are given in relation to a specific action taken by a subject. A positive reward, known as a reinforcer, tries to increase the likelihood of that action, while a negative reward, known as a punishment, tries to decrease the inclination to choose that action (Wahome 2022). By using reinforcers and punishments, the subject's decisions can change with the desire to receive positive rewards and avoid negative rewards.

Deep RL algorithms use a similar system of positive and negative rewards to learn and train. A program, often referred to as an Agent, does so by making decisions through a feedback loop. In Figure 1, the feedback loop is modeled (Medium 2021). As explained by Cody Marie Wild, an Agent is provided with an environment to interact along with a range of discrete or continuous actions it can take in that environment. The environment can range from 3D image layouts to matrixes of numbers, and actions include changing positioning or manipulating values in the environment. To start, the program is given a state, or observation of the environment at that given timestamp. The state is passed to the Agent as its input. After processing the input and passing the values through the network's hidden layers, the network makes an output action, for example, changing its x or y position on a map or multiplying an array of numbers. Using numerous reward functions, a reward is calculated depending on the action taken and the relationship to the end goal (Wild 2018). Depending on the output result, an Agent can earn a reward for an output close to the desired behavior, while movements straying from that will result in a negative reward. Through many attempts of action and reward, the neural network must learn how to maximize the net reward through “trial and error” (Hasemi-Pour). With actions made in the environment, the reward and the new state are passed back to the Agent, and the process repeats. One loop is considered a timestamp, while a group of timestamps is called an episode. A section of an episode is a trajectory (Wild 2018). Like supervised learning, a model is provided a goal to work towards and a function for improvements, however, “once these parameters are set, the algorithm operates on its own, making it more self-directed than supervised learning algorithms” (Hasemi-Pour). A program makes its training data through experience, and some randomness is used to explore the range of output action possibilities.

Figure 1: Model of Reinforcement Cycle



"Reinforcement Learning with PPO." *Medium*, 5 Oct. 2021, odsc.medium.com/reinforcement-learning-with-ppo-6a46e79a8359. Accessed 23 Nov. 2023.

According to Xander Steenbrugge, a program’s terminal state dictates when to end the program, either maxing out on a pre-set time parameter, fulfilling the required reward goal, or lacking any alternative actions to make, essentially being out of options or choices. Sometimes, the program stops with a different end behavior result because of a flawed reward function that does not accurately prioritize the right actions. Programs may take advantage of a certain reward action that is given high rewards but is not as key to performing the end task, repeating the unnecessary action to maximize the reward, but in the end, the action is not desirable for the programmer (Steenbrugge 2018).

An Agent uses a Critic and Actor to maximize the reward it recieves. In PPO, the compiled trajectories from the training set are analyzed, and actions are made by predicting which output will maximize the net reward for the model by looking not only at the present but also at future timestamps (Medium 2021). Further explained by a video produced on B2Studio, a YouTube page that produces and explains creative AI projects replicating human behavior, in PPO, the job is split by multiple neural network models. The first is the Critic, whose role is to judge the environment and determine what true reward is by looking at gathered trajectory data. The second is an Actor interacting with the environment and taking actions to maximize the Critic's estimations. A Critic trains using backpropagation and compares its reward predictions to the actual rewards received. As a comparison, the Actor is like a soccer player, while the Critic is like the coach. Both are trying to win the game through the coach providing strategies and the soccer player following them through.

While assessing choices, the Critic must analyze the entire action sequence to determine if a reward is worth getting. By looking a few timestamps ahead, a Critic wants to avoid performing an action that receives a small momentary positive reward, but net negative rewards as a result of that action, like jumping to get a few coins in a game, but falling off a cliff as a result. To calculate the overall reward return from a decision, the Critic adds all future rewards following the action. The value calculated provides the return on action taken.

However, because the rewards are given at a future instance or timestamp, the value of the reward diminishes depending on how far into the future it is. The reward is not guaranteed to be given since the future holds a level of unpredictability. The concept is modeled by a discount factor, or the rate at which a reward diminishes over time, a number ranging from zero to one. The factor is multiplied to future rewards exponentially, and the new sum of rewards is called the discounted return. Because the Critic is unable to predict randomness used to ensure the actions taken by the program explore all possibilities, the Critic averages all possible values from actions the Actor can take. The calculation is used using the value function, trying to predict the reward received without randomness. The value creates a baseline used to assess the model's performance improvement. To determine successful changes, the Critic uses the advantage function and subtracts the baseline value from the discounted return. A positive number means the randomness improves the behavior, while a negative number signals to the Actor that the behavior is discouraged. Values for the Actor are calculated by multiplying each timestamp by a fraction representing the probability of that decision chosen by the Actor. These numbers are fed into the Actor to make a decision. After multiple rounds and the use of backpropagation, the Actor learns what actions to pick. Training sessions repeat and continue, but when a new training section is started, the episodes from the previous are removed from memory. Once training is complete, the Critic can be removed, and the Actor performs as the Agent by itself (B2studio 2023).

PPO's main advantage and specialty is its ability to limit or "clamp" the range of changes made to model behavior. According to John Schulman, a large problem with most RL algorithms is that sometimes the model behavior changes too much in the wrong direction if given a large reward, ruining previous hard work of training as previous training cases are not saved to correct the changes. The changes can create a model whose behavior is ruined by one case. Because the program changes behavior on trajectories and not on individual action, instead of differentiating a negative reward with the new specific action it makes, the program may generalize it to the entire sequence of actions. PPO solves this by ensuring changes to the model behavior are not drastic through a clipping object function, limiting the change range (Schulman 2017). While PPO can lower the required test cases, it requires an accurate clipping ratio: If too small, the program will learn too slowly, and if too big, it could destabilize the learning (LinkedIn).

A lab implemented PPO on a robotic arm to teach it how to pick up and manipulate objects of different shapes through a virtual environment (Shahid *et al* 2020). Outside of robotics, other applications include Agents learning games such as Pac-Man or other Atari Games (Hashemi-Pour). With PPO's release to the public in 2017, the deep learning algorithm has helped improve robotic and program action training, pushing the possibilities of Neural Network learning.

Works Cited

A. A. Shahid, L. Roveda, D. Piga and F. Braghin, "Learning Continuous Control Actions for Robotic Grasping with Reinforcement Learning," 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Toronto, ON, Canada, 2020, pp. 4066-4072, doi: 10.1109/SMC42975.2020.9282951.

"AI Learns To Swing Like Spiderman." *YouTube*, uploaded by B2studios, www.youtube.com/watch?v=Y48Vk77MoYg&ab_channel=b2studios. Accessed 23 Nov. 2023.

Hasemi-Pour, Cameron. "reinforcement learning." *TechTarget*, www.techtarget.com/searchenterpriseai/definition/reinforcement-learning#:~:text=Reinforcement%20learning%20is%20a%20machine,learn%20through%20trial%20and%20error. Accessed 23 Nov. 2023.

"An introduction to Policy Gradient methods - Deep Reinforcement Learning." *YouTube*, uploaded by Xander Steenbrugge, www.youtube.com/watch?v=5P7I-xPq8u8&ab_channel=ArxivInsights. Accessed 23 Nov. 2023.

Medium. 7 Feb. 2021, towardsdatascience.com/a-graphic-guide-to-implementing-ppo-for-atari-games-5740ccbe3fbc. Accessed 23 Nov. 2023.

"Reinforcement Learning with PPO." *Medium*, 5 Oct. 2021, odsc.medium.com/reinforcement-learning-with-ppo-6a46e79a8359. Accessed 23 Nov. 2023.

Schulman, John, et al. "Proximal Policy Optimization." *OpenAI*, 20 July 2017, openai.com/research/openai-baselines-ppo. Accessed 23 Nov. 2023.

Wahome, Cyrus. "What Is Operant Conditioning?" *WebMD*, 27 Apr. 2022, www.webmd.com/mental-health/what-is-operant-conditioning. Accessed 23 Nov. 2023.

"What are the advantages and disadvantages of PPO compared to other policy gradient methods?" *LinkedIn*, edited by AI and LinkedIn community, www.linkedin.com/advice/1/what-advantages-disadvantages-ppo-compared. Accessed 23 Nov. 2023.

Wild, Cody Marie. "The Pursuit of (Robotic) Happiness: How TRPO and PPO Stabilize Policy Gradient Methods." *The Pursuit of (Robotic) Happiness: How TRPO and PPO Stabilize Policy Gradient Methods*, Medium, 8 July 2018, towardsdatascience.com/the-pursuit-of-robotic-happiness-how-trpo-and-ppo-stabilize-policy-gradient-methods-545784094e3b. Accessed 23 Nov. 2023.