**Proximal Policy Optimization (PPO)**

## Why PPO?

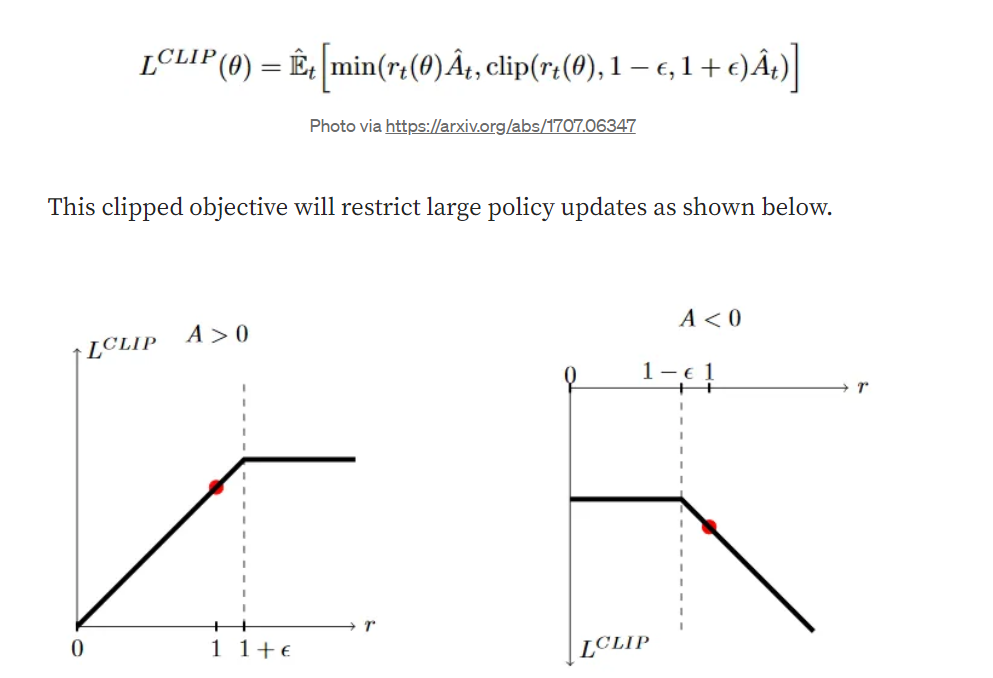
1. **Unstable Policy Update**: In Many Policy Gradient Methods, policy updates are unstable because of larger step size, which leads to bad policy updates and when this new bad policy is used for learning then it leads to even worse policy. And if steps are small then it leads to slower learning.
2. **Data Inefficiency**: Many learning methods learn from current experience and discard the experiences after gradient updates. This makes the learning process slow as a neural net takes lots of data to learn.

PPO comes handy to overcome the above issues.

## Core Idea Behind PPO

In earlier Policy gradient methods, the objective function was something like **LPG(θ) =ˆEt[logπθ(at|st)ˆAt].**But now instead of the log of current policy, we will be taking the ratio of current policy and old policy.

We will be also clipping the ratio and will the minimum of the two i.e b/w clipped and unclipped.



So, the final objective contains 3 parts, first one is Lclip, the second one is MSE of our critic net i.e squared loss of state values predicted and target. And the third part is entropy to encourage exploration.

# Algorithm Steps

1. Play game for n steps and store state, action probability, rewards, done variables.
2. Apply the Generalized Advantage Estimation method on the above experience.
3. Train neural networks for some epochs by calculating their respective loss.
4. Test this trained model for “m” episodes.
5. If the average reward of test episodes is larger than the target reward set by you then stop otherwise repeat from step one.