At the end of this exercise you will have implemented a policy gradient algorithm to solve the `CartPole-v1` environment.

1. **Policy Gradient Implementation**

   - Implement the `Policy` network to solve the `CartPole-v1` environment.
   - Implement `compute_returns` to compute the discounted returns G_t for each state in a trajectory.
   - Implement the `policy_improvement` step to update the policy given the rewards and probabilities from the last trajectory.
   - Use the policy in the `act` function to sample an action and return its log probability.

2. **Questions**

   - What could be a problem in the current implementation? How does the length of the trajectories affect the training?
   - How could a baseline be implemented to stabilize the training?
   - Does the same network architecture and learning rate work for `LunarLander-v2`?
   - How is the sample complexity (how many steps it takes to solve the environment) of this algorithm related to the DQN from the last exercise?