You need to help a user to analyze a control policy for the task Pong available in the OpenAI Gym repository. The policy is obtained with deep reinforcement learning.
You need to first understand the goal of the game and the policy.

# Game Description

There are two paddles in the game screen, which are located at the left and right side of the screen. The agent controls the right paddle, and its opponent controls the left paddle. Both of them can only control the paddle to move up or down. They cannot move leftward or rightward

Like a pingpong game, the agent competes against its opponent by stricking the ball to the opponent's side (left). The agent earns a point if its opponent fails to strick the ball back.

The agent needs to solve the game in discrete steps. At each step, it takes as input the game screen, and it needs to take one of the three actions:

* noop: take no operation
* up: move its paddle upward.
* down: move its paddle downward.

# The policy

## Input Variable

We set up a xOy-coordinate system for the game screen. The origin is at the upper left corner. The positive direction of the y-axis is downwards, and the positive direction of the x-axis is to the right. We provide the agent with the latest four consecutive frames and use the coordinates of objects in these frames as input. Frame 4 is the current frame. Frame 3 is the frame obtained at one step before. Frame 2 is the frame obtained at two steps before, and frame 1 is the frame obtained at three steps before. You can use the coordinates of the same object in different time steps to infer the motion of the object.

The objects of interest are the agent, the opponent, and the ball. The input variables follows this naming convention: [x/y]_object_frame. For example, x_agent_1 is the x coordinate of the agent at frame 1. Remember, the input variables represent coordinates of some objects, and they are in the range [0,1].

## Logits

logits_noop1 = -0.56*y_agent_1**2 - 0.38*y_agent_1*y_agent_2 - 0.087*y_agent_1*y_opponent_1 - 0.16*y_agent_1*y_opponent_2 - 0.76*y_agent_1*y_opponent_3 - 0.51*y_agent_1*y_opponent_4 - 0.54*y_agent_1 - 0.24*y_agent_2**2 - 0.073*y_agent_2 + 0.27*y_agent_4**2 + 0.55*y_agent_4 - 0.078*y_opponent_1**2 - 0.33*y_opponent_1*y_opponent_2 - 0.2*y_opponent_1 - 0.35*y_opponent_2**2 - 0.5*y_opponent_2 - 0.34*y_opponent_3**2 - 0.45*y_opponent_3*y_opponent_4 - 0.32*y_opponent_3 - 0.15*y_opponent_4**2 - 0.19*y_opponent_4 + 1.1

logits_noop2 = -0.074*y_agent_1*y_opponent_2 + 0.059*y_agent_1*y_opponent_3 - 0.097*y_agent_4 - 0.16*y_opponent_1*y_opponent_2 - 0.18*y_opponent_2**2 - 0.27*y_opponent_2 + 0.063*y_opponent_4

logits_up1 = 0.23*y_agent_1**2 + 0.59*y_agent_1*y_agent_2 + 0.4*y_agent_2**2 + 0.11*y_agent_2 - 1.5*y_agent_4**2 - 3.6*y_agent_4 + 0.068*y_opponent_3 + 1.1

logits_down1 = 0.09*x_ball_3 + 0.12*x_ball_4 - 0.21*y_agent_1**2 + 0.12*y_agent_1*y_opponent_1 + 0.27*y_agent_1*y_opponent_2 - 0.43*y_agent_1*y_opponent_3 - 0.28*y_agent_1*y_opponent_4 + 0.13*y_agent_2 + 0.14*y_agent_4**2 + 0.43*y_agent_4 + 0.087*y_ball_3 + 0.15*y_ball_4 + 0.14*y_opponent_1**2 + 0.6*y_opponent_1*y_opponent_2 + 0.61*y_opponent_1 + 0.65*y_opponent_2**2 + 1.1*y_opponent_2 - 0.2*y_opponent_3**2 - 0.26*y_opponent_3*y_opponent_4 - 2.8*y_opponent_3 - 0.085*y_opponent_4**2 - 0.14*y_opponent_4 - 2.3

logits_up2 = 0.063*x_ball_4 - 0.078*y_agent_1 + 0.18*y_agent_2**2 + 0.52*y_agent_2*y_agent_3 + 0.35*y_agent_2*y_opponent_1 + 0.29*y_agent_2*y_opponent_2 + 0.26*y_agent_2 + 0.38*y_agent_3**2 + 0.51*y_agent_3*y_opponent_1 + 0.42*y_agent_3*y_opponent_2 + 1.6*y_agent_3 - 8.2*y_agent_4 - 0.085*y_ball_3 + 0.17*y_opponent_1**2 + 0.28*y_opponent_1*y_opponent_2 + 0.25*y_opponent_1 + 0.11*y_opponent_2**2 + 0.15*y_opponent_2 - 0.074*y_opponent_3 + 0.26

logits_down2 = -0.052*x_ball_1 - 0.068*x_ball_3 - 0.093*x_ball_4 + 0.18*y_agent_1 - 0.17*y_agent_2**2 - 0.49*y_agent_2*y_agent_3 - 0.33*y_agent_2*y_opponent_1 - 0.27*y_agent_2*y_opponent_2 - 0.39*y_agent_2 - 0.35*y_agent_3**2 - 0.48*y_agent_3*y_opponent_1 - 0.4*y_agent_3*y_opponent_2 - 0.38*y_agent_3 + 0.15*y_agent_4**2 + 0.54*y_agent_4 - 0.06*y_ball_1 - 0.064*y_ball_3 - 0.11*y_ball_4 - 0.17*y_opponent_1**2 - 0.28*y_opponent_1*y_opponent_2 - 0.58*y_opponent_1 - 0.13*y_opponent_2**2 - 0.38*y_opponent_2 + 2.2*y_opponent_3 - 0.052*y_opponent_4 - 3.6

## The Probability of Actions

action_noop = [exp(logits_noop1) + exp(logits_noop2)] / sum(exp(logits))

action_up = [exp(logits_up1) + exp(logits_up2)] / sum(exp(logits))

action_down = [exp(logits_down1) + exp(logits_down2)] / sum(exp(logits))