ABUDUREHEMAN ADILA
SID: 916186665
WINTER 2020
ECS170

LEARNING TO PLAY PONG REPORT

PART I - Problem Representation

1. Hardware Ram as array would be observing the RGB colors of the each pixel in the image. And Hardware ram array would be observing the RAM bytes. Both cases each action is repeated based on the frames. Both are trying to get the best representation of the game states and what we want is try to train our inner layers with input layers and get the most correct action with output layers. However, I believe the RAM is a easier to learn because it has info about the direction of the ball or it can help to avoid the task of learning a 'representation' of the game, and that let's you move directly to learning a 'policy' or deciding what the next move is.

2. The NN takes a frame image of the game as the input, where each pixel corresponds to an individual input as neuron, and outputs a state action pairs to Q values. We can train a neural network on samples form the state action to learn the actions. Like all neural networks, they use coefficients to approximate the function relating inputs to outputs, and they are learning consists to finding the right weights by iteratively adjusting those weights along gradients that promise less error. In reinforcement learning, convolutional networks can be used to recognize an agent’s state when the input is image pixels where represents a state, a convolutional net can rank the actions possible to perform in that state as rewards of state action.

self.batch_size	group of training samples processed before the model is updated.
self.gamma	discount factor for reward
self.num_frames	The total number of frames for the training model to be trained.
self.replay_buffer	this buffer records past states, the actions taken at those states, the reward received and the next state that was observed.
self.env	The game environment
self.input_shape	the matrix shape of the image or frame.

self.num_actions	number of possible actions for the pong.
self.features	Extracted the important features from the input state with image pixels.
self.fc	it returns the output of Q values of the input state, where its indices are the possible actions.
def forward	get the Q values or output of the passed input state.
def act	get the best action move (return index of the MAX reward among the Q values of the input state) or a random move to explore other options.

3. It represents the epsilon greedy algorithm where agent decides to explore or exploit based on the value of epsilon. Exploring would try out the random values and exploiting would be taking the best action was last seen with the highest Q value. So when an action is selected in training, it will be either get the best value where the highest Q value or a random action to to look for a more optimal solutions.

PART II - Making the Q-Learner Learn The objective function is to steer the current Q value towards the target Q value by maximizing the decrease in error.

S represents the next possible state
A represents the action taken
Q represents the target model
Loss represents the mean square error between the predicted and actual values
The Θ_i represents the model parameters at the ith iteration, and the yi represents the current Q of its iteration. $Q(s, a; \Theta_i)$ represents the next Q values calculated from the given model parameters. The current Q values are subtracted from calculated Q and used the mean share error.

PART IV - Learning to Play Pong

The rewards are increasing as the #of frame increases and the loss is between 0 to 0.078.