DQN for Training Open AI Gym Games ashish | March 27, 2020 **Start** Initialialisation **Initialise Hyperparams** (Target Update as C) Policy model with random weights Make the Atari **Target model with Policy Environment of** model weights your choice Replay buffer with M capacity (MsPacman) number of episodes = N -Next i-For i in 0..N-1 Exit Loop and Close the Environment Stop Next i Enter loop \blacksquare Target model Νo Weights = Policy **Initialise Environment** model weights Initialise State = current screen - last screen Yes For t in count() steps or number of Actions taken If i = C ε-Greedy Action selection if probability>ε Yes Νο Take step based on the Ν'n Reward (saved in an array), Done (true if episode completes) Yes Update the current If Done If Done -No-Yes State = Current_Screen -Last_Screen Sample random Minibatch of Transitions from Replay Buffer Obtain Q values from Policy model **Optimize Model** Obtain expected Q values from Target Model Perform gradient descent step on Huber loss and update model weights