Deep Q Learning From Demonstrations (a.k.a. Learning From Videos)

Link: <https://arxiv.org/pdf/1704.03732.pdf>

Basic equations:

Basic Requirements:

* We need to pretrain on demonstration data using a combination of temporal difference (TD) and supervised losses
* Supervised loss tells it to imitate the demonstrator
* TD loss tells it to learn a consistent relationship that can be continued via RL
* The ratio of pretraining and pure deep q learning is critical
* Tau states how often the target network should receive the normal q network’s weights
* Prioritized experience replay can be added
* We will need a DQN and Target Q Network
* We will need a replay buffer (a.k.a. memory)

The Model:

Goal: we want the agent to learn as much as possible from the demonstration data before its trained in the actual environment

1. The model samples batches of the video data
2. Then applies four losses: 1 step Q Learning Loss, n-step Double Q Learning Loss, supervised large margin classification loss, and L2 regularization loss on the weights and biases
3. Overall loss is then calculated by adding all the 4 losses

* Supervised large margin classification loss: used to classify the actions
* The Q Learning losses make sure that the network satisfies the Bellman equation (the equation that clusters states, actions, and rewards so that Q Learning can see how a state and action can output a certain reward)
* The Target Q Network calculates the overall loss
* N-step loss takes in all the rewards up to a point and returns the best state and action pair
* The agent NEVER overwrites demonstration data. Therefore, we may need 2 replay buffers

Training loop steps:

1. Initialize one replay buffer with video dataset
2. Initialize random weights for the model
3. Initialize the frequency that we want to update the Target Q Network
4. Initialize the number of epochs we want to train the model
5. Train the main model and calculate the loss using the Target Q Network
6. Update the weights of the model using the gradients
7. Train the agent on the environment
8. Calculate the overall loss using the Target Q Network
9. Update the weights of the model using the gradients