-------------------------------------------------------------------------------------------------------------------------------

**Question 1. (150 words, 2 points)** What is Experience Replay and how did the DeepMind team implement it in DQN? What are the advantages and disadvantages to using Experience Replay?

**Answer (273 words):** Experience Replay is a biologically inspired mechanism that randomizes over the data, thereby removing correlations in the observation sequence and smoothing over changes in the data distribution.

Experience Replay is implemented by storing the agent's experience at each time step in a dataset . During learning, Q-learning updates are applied on samples of experience , drawn uniformly at random from the pool of stored examples.

Advantages of using Experience Replay include: 1). Each step of experience is potentially used in many weight updates, which allows for greater data efficiency. 2). Learning directly from consecutive samples is inefficient, owing to the strong correlations between the samples, randomizing the samples breaks these correlations and therefore reduces the variance of the updates. 3). When learning on-policy the current parameters determine the next data sample that the parameters are trained on. With Experience Replay the behavior distribution is averaged over many of its previous states, smoothing out learning and avoiding oscillations or divergence in the parameters.

Disadvantages of using Experience Replay include: 1). Some important transitions are delayed to make effect by Experience Replay, this negative effect is partially controlled by the size of memory buffer. The algorithm presented in the paper only stores the last N experience tuples in the replay memory, and samples uniformly at random from when performing updates. This approach is in some respects limited because the memory buffer doesn’t differentiate important transitions and always overwrites with recent transitions owing to the finite memory size . 2). Multi-step algorithms such as , when well-tuned, can improve learning curves. With Experience Replay, it is harder to use multi-step algorithms.

-------------------------------------------------------------------------------------------------------------------------------

**Question 2. (150 words, 2 points)** Explain what Figure 4 represents and its significance.

**Answer (193 words):** Figure 4 represents the last hidden layer assigned by DQN to game states experienced while playing Space Invaders. The points are colored according to the state values (V, maximum expected reward of a state) predicted by DQN for the corresponding game states (ranging from dark red (highest V) to dark blue (lowest V)). Besides, both full (top right screenshots) and nearly complete (bottom left screenshots) screens are predicted with high state values by the DQN agent, because it has learned that completing a screen leads to a new screen full of enemy ships.

The significance of Figure 4 lies in the fact that the t-SNE algorithm tends to map the DQN representation of perceptually similar states to nearby points. Additionally, the fact that DQN representations of states that are close in terms of expected reward but perceptually dissimilar appear to have similar embeddings generated by the t-SNE algorithm is consistent with the notion that the network is able to learn representations that support adaptive behavior from high-dimensional sensory inputs. Furthermore, it can be shown that the representations learned by DQN are able to generalize to data generated from policies other than its own.

---------------------------------------------------------------------------------------------------

**Question 3. (250 words, 2 points)** In algorithm 1, what is the motivation behind choosing a random action with probability e? Why not just choose the action with the most expected value? Why does the probability e decrease over time? Propose another possible scheme for choosing an action besides choosing a random action that would also satisfy this motivation and how you might implement it.

**Answer (295 words):** The motivation behind choosing a random action with probability is that this ensures adequate exploration of the state space. In practice, the behavior distribution is selected by an -greedy policy that follows the greedy policy with probability and selects a random action with probability . Also, a deterministic policy may get stuck in a local optimal, whereas this is less likely with a non-deterministic policy.

Choosing always the action with the most expected value results in insufficient exploration of possible actions especially at the beginning.

starts with some large value and decreases over time. This way, at the beginning of the training, an action is uniformly selected among all possible actions, and as the training progresses, the action with the most expected value is selected more and more frequently. Essentially, this decrease over time allows maximum exploration at the beginning and exploitation as time goes by.

Besides choosing a random action, another classic approach is the so-called entropy regularization, which is used in policy gradient methods when we’re adding entropy of the policy to the loss function, punishing the model for being too certain in its actions. Both -greedy and entropy regularization need to be adjusted to the environment, and both don’t take into account the current situation experienced by the agent.

Another possible scheme for choosing an action that would also satisfy this motivation is the so-called Noisy Nets. The idea is to add parametric noise to the fully connected layer to aid exploration. Parameters of the noise can be adjusted by the model during training, allowing the agent to decide when and in what proportion it wants to introduce the uncertainty in weights. Two implementations of the parametric noise are possible, namely Independent Gaussian Noise, and Factorized Gaussian Noise.

**Reference:**

shorturl.at/cixP0

shorturl.at/gmwDZ

---------------------------------------------------------------------------------------------------

**Question 4. (300 words, 5 points)** Research and explain one of the following alternative architecture that improves on DQN: Deep Attention Recurrent Q Network, **Dueling Architectures for Q-Learning**, Deep Double Q-Learning Network. You should read one paper on the topic of your choice, summarize the paper, and indicate how it improves on DQN. Please cite the paper you refer to.

**Answer (357 words):** The dueling architecture consists of two streams that represent the value and advantage functions, while sharing a common convolutional feature learning module. The two streams are combined via a special aggregating layer to produce an estimate of the state-action value function .

The dueling architecture can learn which states are (or are not) valuable, without having to learn the effect of each action for each state. This can be useful in states where its actions do not affect the environment in any relevant way.

The dueling architecture can be beneficial since the agent may not need to care about both value and advantage at the same time. Therefore, a more robust estimate of the state value can be achieved by decoupling it from the necessity of being attached to specific actions. In fact, experiments demonstrate that the dueling architecture can identify the correct action more quickly during policy evaluations as redundant or similar actions are added to the learning problem.

The key insight behind the dueling architecture is that for many states, it is unnecessary to estimate the value of each action choice. In some states, being able to know which action to take is extremely important, whereas in many other states this is not the case since the choice of action has no influence on what happens next.

Comparing a single-stream architecture with the dueling architecture with relatively large number of actions, the dueling architecture performs better than the traditional -network. In the dueling architecture, the value stream learns a general value that is shared across many similar actions, leading to faster convergence.

In summary, the dueling architecture is advantageous in that it is able to learn the state-value function efficiently. In contrast to the updates in the single-stream architecture where only the value for one of the actions is updated and the values for all other actions remain untouched, in the dueling architecture, with every update of the values, the value stream is updated. This allocates more resources to , and thus allows for better approximation of the state values, which in turn need to be accurate for -learning to work.

**Reference:**

<https://arxiv.org/pdf/1511.06581.pdf>

shorturl.at/jzDKX