Experience replay is a key technique in RL in which the agent repeatedly learns on previous experiences stored in a buffer in order to improve sample efficiency. Prioritized Experience Replay (PER) weights experiences by their TD errors to prioritize important experiences during replay. We propose Cluster-wise Learnability Experience Replay (CLER) that prioritizes experiences based on their learnability approximated by cluster-wise regression on TD error. Results suggest further improved sample efficiency.

Goal: Develop a prioritization method for experience replay to improve sample efficiency.

**Prior work on ER**

**Experience Replay**

- For $k = 1, \ldots, m$:
  - Observe $(s, a, r, s')$, place in buffer
  - For $j = 1, \ldots, M$:
    - Uniformly sample transitions
    - Accumulate weight change
    - Update weights
    - Update target network

**Prioritized Experience Replay**

- For $k = 1, \ldots, m$:
  - Observe $(s, a, r, s')$, place in buffer with maximal priority
  - For $j = 1, \ldots, M$:
    - Uniformly sample transitions from buffer according to weighted probabilities
    - Compute TD errors
    - Set probabilities to TD error
    - Accumulate weight change
    - Update weights
    - Update target network

**Our prioritization algorithm**

Cluster-wise Learnability Experience Replay (CLER)

- For $k = 1, \ldots, m$:
  - Observe $(s, a, r, s')$, place in buffer
  - Cluster buffer using mean shift
  - For $j = 1, \ldots, M$:
    - Sample transition from buffer according to weighted probabilities
    - Compute TD errors
    - For $i = 1, \ldots, m$:
      - Linearly regress on TD errors in the $i$-th cluster to obtain slope $\beta$
      - Set probabilities of all experiences in the $i$-th cluster to $\sigma(\beta)$ (learnability approx.)
    - Accumulate weight change
    - Update weights
    - Update target network

**Mean shift** is a clustering algorithm that uses a sliding window to compute new centroids and can remove duplicate centroids.

**CLER (step by step)**

- Insert New Experiences Into Buffer
- Cluster Replay Buffer
- Linearly Regress on TD Error by Cluster
- Assign $\sigma(\beta)$ as Priority of Cluster

**Task / Environment:**

We modified CartPole-v0 from OpenAI Gym such that at each frame, with probability $\epsilon$, a random action is taken to create noisy states.

**Policy / Architecture:**

We use Q-learning with epsilon-greedy exploration. Our Q-function is a two-layer fully connected network with 128 hidden units.

**Discussion:**

- Compared to PER, our learnability-based CLER algorithm obtains significantly higher test time reward at about 11% fewer environment interactions, showing lower sample complexity.
- We hypothesize that CLER performs better because it is able to ignore states with high TD error and low learnability, such as when the pole will definitely fall.

**Future directions:**

- Better ways to approximate learnability and speed up CLER
- Explore probabilistic and theoretical guarantees on performance
- More rigorous hyperparameter tuning e.g. number of clusters, clustering method

**Works Cited**