Paper Critique

Shuvrajeet Das, DA24D402

Course: DA7400, Fall 2024, IITM
Paper: [Vector Quantized Models for Planning]
Date: [23-12-2024]

Make sure your critique Address the following points:
1. The problem the paper is trying to address
2. Key contributions of the paper
3. Proposed algorithm/framework
4. How the proposed algorithm addressed the described problem
Note: Be concise with your explanations. Unnecessary verbosity will be penalized. Please don't exceed 2 pages.

---

1 The problem the paper is trying to address

The paper addresses the challenge of planning in **stochastic** and **partially observable** environments for **reinforcement learning** (RL). Traditional methods like *MuZero* perform well in deterministic, fully observable settings but face difficulties in handling stochastic dynamics and incomplete information. The key idea proposed in this paper is to use *Vector Quantized Variational AutoEncoders (VQVAE)* to discretize latent variables of states and actions, enabling efficient planning using *Monte Carlo Tree Search (MCTS)* in these complex environments.

- **Problem**: Planning in environments that are **stochastic** and **partially observable**.
- **Solution**: Employing *VQVAE* to represent the environment's latent variables for planning with *MCTS*.

This approach allows planning over both **agent actions** and **environment responses**, which improves performance in stochastic and partially observable environments.

2 Key contributions of the paper

- Proposing the use of *Vector Quantized Variational AutoEncoders (VQVAE)* to encode both the state and action space into discrete latent variables, which facilitates planning in **stochastic** and **partially observable** environments.
- Introducing a **hybrid** model that allows planning over both **agent actions** and **environment responses** using *Monte Carlo Tree Search (MCTS)* in the discrete latent space.
- Developing a framework that performs well in both *offline reinforcement learning* settings and *large visual observation spaces*, demonstrating scalability.
- Showing that the proposed method outperforms the *MuZero* algorithm in a stochastic version of chess and in the *DeepMind Lab* environment, which includes **large, partially observable, and stochastic** visual inputs.

3 Proposed algorithm/framework

The proposed framework consists of the following components:

- **State VQVAE**: A *Vector Quantized Variational AutoEncoder (VQVAE)* is trained to encode the sequence of states and actions into discrete latent variables. This involves an encoder-decoder pair:
  - Encoder: Compresses the sequence of states and actions into a discrete latent variable.
  - Decoder: Reconstructs the state sequence from the latent variables.
- **Transition Model**: A transition model is trained using the discrete latent variables. The model alternates between predicting *agent actions* and *environment latent variables* during planning. The transition model is used to simulate future trajectories in the latent space.
- **Monte Carlo Tree Search (MCTS)**: The *MCTS* algorithm is used for planning over both **actions** and **latent variables**. The MCTS search tree has two types of nodes:
  - **Action nodes**: Predict the next action using a policy learned from the transition model.
  - **Stochastic nodes**: Predict the next discrete latent state (environment response) using a policy over the latent variables.
- **VQHybrid and VQPure Variants**: Two variants of the planning path are proposed:
  - **VQHybrid**: Alternates between actions and latent variables in the planning path.
  - **VQPure**: Plans directly in the discrete latent space by encoding both actions and environment responses into latent variables.
- **Training Process**: The model is trained in two stages:
  1. **Stage 1**: Train the *State VQVAE* on sequences of states and actions.
  2. **Stage 2**: Train the *Transition Model* using the discrete latent variables generated by the State VQVAE.
- **Loss Function**: The total loss combines

4 How the proposed algorithm addressed the problem

- **Discrete Latent Representations**: Using *VQVAE*, the algorithm encodes continuous state and action spaces into **discrete latent variables**, enabling efficient planning in uncertain or partially observable environments.
- **Handling Stochasticity**: It models **stochastic responses** by planning over latent variables, accounting for uncertainty in the environment.
- **Monte Carlo Tree Search (MCTS)**: The framework extends MCTS to plan over both actions and latent variables, incorporating environment uncertainty during search.
- **Partial Observability**: The algorithm generalizes to **partially observable environments**, planning in latent space without needing full state access.
- **Efficiency and Scalability**: By discretizing actions and states, the algorithm reduces complexity, making it scalable to large environments like *DeepMind Lab*.