VIPER[1] Summary

Erik Schwede

December 2024

1 Problem Statement

Reinforcement learning usually requires properly constructed reward functions, which are typically handcrafted and demand some expert knowledge.

2 Related Work

Other papers primarily employ three main approaches to address the problem statement.

2.1 Labelling videos with predicted actions

This approach employs an inverse dynamics model trained to automatically label frame transitions of expert videos with actions. To achieve this, the inverse dynamics model must first be trained using manually labeled ground truth data. The downside of this approach is its heavy reliance on ground truth labels, as well as the quantity and diversity of the training data.

2.2 Reinforcement learning with videos

Several approaches exist: some provide rewards based on the estimated progress of the task, while others reward based on the differences between the expert and agent trajectories. Another common approach involves using a discriminator that competes with the agent. The discriminator attempts to distinguish between expert videos and the agent's gameplay, while the agent tries to fool the discriminator. However, this approach is often error-prone, as the discriminator tends to focus on irrelevant features, such as background color.

2.3 Using video models as policies

Text-conditioned video generation models are used to plan a trajectory, which is then executed using an inverse dynamics model. However, this process can be quite slow, as the video must be generated first.

3 Key Contribution

The authors propose VIPER (Video Prediction Rewards), a system that leverages video prediction models trained on expert videos to provide rewards, instead of relying on environment rewards. The reward signal is based on the likelihood of the agent's trajectory under the pretrained video prediction model. Additionally, an exploration term is incorporated to encourage diversity, leading to the following reward formulation:

$$r_t^{\text{KL}} = r_t^{\text{VIPER}} + \beta \cdot r_t^{\text{expl}}$$

4 Methodology

4.1 Video Prediction Model

First, a VQ-GAN is trained to encode video frames into latent codes. In the paper, 64x64 images are used, and the VQ-GAN encodes them into 8x8 or 16x16 discrete codes, depending on the visual complexity. Second, an autoregressive transformer, based on VideoGPT, is employed to model the distribution of codes using a maximum likelihood objective:

$$\max_{\theta} \sum_{t=1}^{T} \sum_{i=1}^{Z} \log p_\theta \left(z_t^i \mid z_t^{1:i-1}, z_{1:t-1} \right)$$

T : Total number of frames in the sequence
Z : Number of discrete latent codes per frame
p_θ : Sequence distribution of the video model
z_t^i : The i-th latent code at the t-th frame

4.2 Reward Formulation

The reward is defined as the log-likelihood of the agent's transition (x_t, a_t, x_{t+1}) under the video model:

$$r_t^{VIPER} = \ln p_\theta(x_{t+1} \mid x_{1:t})$$

Additionally, an exploration bonus is included to encourage the agent to explore new trajectories:

$$r_t^{\text{expl}} = \text{H}[q(x_{t+1} \mid x_{1:t})]$$

q : Agent's trajectory distribution

This results in the following reward formulation:

$$r_t^{\text{KL}} = r_t^{\text{VIPER}} + \beta \cdot r_t^{\text{expl}}$$

β : Weight of the exploration bonus

This result in a reward that is maximal when the KL-divergence between $q(x_{1:T})$ and $p(x_{1:T})$ is minimal.

5 Experiments

To evaluate VIPER, the authors used the DeepMind Control Suite for continuous control tasks, Atari games for discrete control tasks, and RLBench for robotic manipulation tasks. For Atari games, the rewards were less noisy when the scoreboard was masked out, as comparing the full image could lead to out-of-distribution scores, resulting in lower rewards.

In the case of RLBench, VIPER even outperformed the Task Oracle because RLBench provides very sparse rewards after long sequences, whereas VIPER delivers dense rewards.

The results demonstrate that VIPER achieves performance comparable to experts and clearly outperforms other approaches, such as inverse dynamics models or adversarial imitation learning.

5.1 Ablation Studies

There are three key design decisions that can be adjusted to achieve better performance.

5.1.1 Exploration objective

The authors use the Plan2Explore reward, which provides a proportional reward based on the disagreement among an ensemble of one-step dynamics models. They observed that the absence of an exploration objective results in the collapse of the policy's behavior. Conversely, increasing the exploration weight β leads to improved performance.

5.1.2 Video model

VIPER is compatible with any video model capable of computing conditional frame likelihoods. The authors also tested MaskGIT, BYOL, and VideoGPT. Among these, VideoGPT outperformed the others, and thus it was selected for use in their experiments.

5.1.3 Context length

The context length refers to the number of frames the model considers at once. In other words, it determines how much of the history is taken into account

to predict the likelihood of the next frame. Experiments have shown that increasing the context length can improve performance, depending on the type of task.

6 Conclusion

VIPER is a general algorithm that provides rewards for reinforcement learning agents by estimating the likelihood of a video frame generated as a result of an agent's action. These likelihoods are estimated using a model trained on expert videos.

One downside of VIPER is the requirement for expert videos to train the video model. If the training data contains uncertainties, such as unpredictable events or inconsistencies in the expert videos, it could lead to higher rewards for states that minimize the model's uncertainty but may not represent the optimal path.

References

[1] A. Escontrela *et al.*, *Video prediction models as rewards for reinforcement learning*, 2023. arXiv: `2305.14343 [cs.LG]`. [Online]. Available: `https://arxiv.org/abs/2305.14343` (cit. on p. 1).