Imagination-Augmented Agents for Deep Reinforcement Learning

Theophane Weber, Sebastien Racaniere, David P. Reichert, Lars Buesing, Arthur Guez, Danilo Rezende, Nicolas Heess, Yujia Li, Demis Hassabis, Adria Puigdomenech Badia, Oriol Vinyals Razvan Pascanu, Peter Battaglia, David Silver Daan Wierstra

***Abstract*—The paper combines model free and model based architectures to give an *imagination-augmented* RL agent (I2A) that can learn to interpret environment models and augment it with model free decisions. The authors give results for improved data efficiency, performance, and robustness to model misspecification.**

I. Summary of I2A architecture

Environment Model : It is a recurrent architecture (LSTM) that can be trained (unsupervised) using the agent's trajectories. It takes the current observation (or history of observations) and the current action as inputs and predicts the next state and any number of signals from the environment (Eg : rewards, dead/alive, etc). (Here : output $\implies$ pixel-wise probability distribution for the output image, and a distribution for the reward). Training data for the environment model was generated from trajectories of a partially trained standard model-free agent.

Rollout Policy Network : The next step is predicted by the environment model is based on an action sampled from the rollout policy. The imagination-augmented policy (policy module) is distilled into a model-free policy by adding cross entropy loss between the imagination-augmented policy and the rollout policy on the current observation, to the total loss. It imitates the policy module, making the internal rollouts similar to the trajectories in the real environment, ensuring rollouts correspond to trajectories with high reward and also, the imperfect approximation results in a rollout policy with higher entropy (balancing exploration and exploitation).

Imagination Core : Predicts the next time step conditioned on an action sampled from the rollout policy. (Comprises of both Environmental model and Rollout Policy network). It is used to produce n trajectories $\hat{\mathcal{T}}_1, \ldots \hat{\mathcal{T}}_n$ (one rollout for each possible action in the environment; subsequent actions are produced by the rollout policy).

Rollout Encoder : It learns to interpret the imagined rollout. It extracts useful information from the rollout relevant for making the current decision (can even ignore the rollout if it is unnecessary.

Aggregator : It aggregates the different rollout embeddings (output of the rollout encoders) into a single imagination code c_{ia} (Here : concatenates)

Model-free path : It comprises of standard network of convolutional layers and fully connected layers.

Policy module : The network takes the information c_{ia} from the model-based path and c_{mf} from the output of the model-free path and outputs imagination-augmented policy vector π and estimated value V. An entropy regularizer is added to the policy to encourage exploration (in addition to the auxiliary loss).

Overall summary : The model based path comprising of the environment model along with the rollout policy are used to simulate imagined trajectories, which are interpreted by a neural network (Rollout Encoder) and provided as additional context (along with the inputs from the model-free path) to a policy network which is responsible for taking actions in the real environment. Since pre-training the environment model led to faster runtime of the I2A architecture, it was used instead of jointly training by adding model loss to the total loss (as an auxiliary loss).

II. Insights

I2A is designed keeping the model imperfection in mind. It learns to combine information from both the model-free and the model-based paths. Without the imagination-augmentation path, I2A becomes a standard model-free network. This method of augmenting model-free agents by providing additional information from model-based planning, was found to possess more expressive power than the baseline model-free agent. Using longer rollouts, while keeping the same number of parameters, increases performance (with higher computational cost) but diminishing returns were observed after a point with longer rollouts.

III. Experiments

The baseline agents (for Sokoban) include **Standard model-free agent** which doesn't posses the model-based path and **Copy-model agent** where the environment model is replaced with a 'copy' model that returns the input observation. But the agent using imagination information from in the environment model rollouts outperform the rest. For the **Rollout encoder-free agent**, high accuracy environment model depicts similar performance as that of I2A, but unlike I2A, its performance degrades catastrophically when using a poor model (susceptible to model misspecification). An environment model predicting only observations and no rewards recovered performance close to that of the original I2A after much longer training. **MCTS variant** requires a nearly high accuracy model and much larger number of rollouts for achieving high performance. For Sokoban, I2A generalises over different number of boxes. In MiniPacman, the model was trained to solve multiple tasks with shared state-transitions. The I2A agent outperformed the standard agents in all tasks, and the copy-model baseline in all but one task