KI DELTA LEARNING
Scalable AI for Automated Driving

Continual Learning for Model-Based Reinforcement Learning

Tim Joseph

Environment / Training Strategies

Use-Case in KI Delta Learning

The main goal is to train an agent in a sequential manner without retraining it every time new data/tasks are available, yet to perform well on all previously seen data/tasks. Simply storing all past data, combining it with new data, and retraining the agent from scratch is not scalable due to memory, computation, and privacy constraints. Additionally, training an agent in a naive way whenever new data is available leads to performance degradation. However, applying a continual-learning approach on our agent we can train it sequentially on multiple tasks without retraining it or observing performance collapse on old tasks.

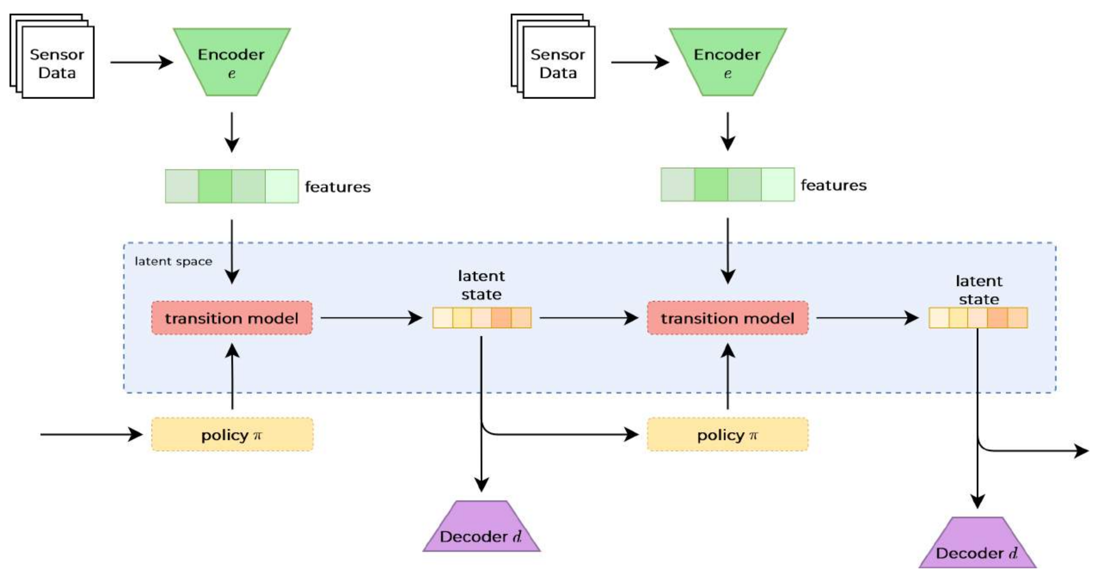

Figure 1: The architecture of our model-based reinforcement learning agent. The encoder, decoder, transition model and policy are regularized independently depending on the delta between tasks.

Problem Motivation

Keeping all collected data indefinitely for training is often infeasible. However, training a model in a naive way whenever new data is available leads to catastrophic forgetting, a phenomenon that describes the abrupt loss of knowledge of previously learned tasks as information relevant to the current task is incorporated.

Problem Solution

Model-based reinforcement learning agents consist of a learned environment model and a policy that is trained based on rollouts of the learned model. We train our agent in a sequential manner to perform well on each task and apply additional penalization to different parts of the architecture depending on the application scenario. For example in the case of a domain-incremental scenario, we penalize only the encoder and decoder, i.e. the part of our neural network that interprets the sensor data, but does not perform any prediction, only the input distribution changes, but the vehicle dynamics do not. In the task-incremental scenario, we penalize all parts of the model and train a separate policy for each task. At the end of each task regularization strengths are determined for every parameter of the neural networks and used in subsequent tasks to prevent forgetting. Every N steps we evaluate the agent's performance on all tasks to analyze the magnitude of forgetting and forward/backward transfer knowledge.

Evaluation

We evaluate our agent in two ways: For performance reasons and to iterate fast over experiments, we use the DeepMind Control Suite environments to validate our general approach.
Then, we further evaluate with the CARLA simulator where we create different sequential tasks, e.g., changing weather, daytime and/or town, as shown in the first row of Figure 2. We analyze the reconstruction capability as well as the predictions of the model. By tracking the performance over multiple tasks we can show, which regularization approach is most compatible with our model-based RL agent.

Figure 2: First row shows a sequence of tasks changing the daytime and weather conditions using the CARLA simulator. The second row shows a sequence of tasks using DMC-benchmarks that were used for benchmarking purposes. The agent is trained in the order in which the images appear.

Partners

External partners

For more information contact:

joseph@fzi.de

KI Delta Learning is a project of the KI Familie. It was initiated and developed by the VDA Leitinitiative autonomous and connected driving and is funded by the Federal Ministry for Economic Affairs and Climate Action.

KI FAMILIE

www.ki-deltalearning.de @KI_Familie KI Familie

Supported by:
Federal Ministry for Economic Affairs and Climate Action
on the basis of a decision by the German Bundestag