Better World-models with Conditional GANs

Nikos Kondylidis, Tycho van der Ouderaa, Alessandra van Ree, Ioanna Sanida
Institute for Informatics, University of Amsterdam

Introduction

We propose a model-based reinforcement learning method, in which we learn a model of the world using a conditional generative adversarial networks.

- ▷ **Biologically motivated**: Humans also develop model of the world [2]
- ▷ **Sample-efficiency**: Less samples from environment by planning
- ▷ **Interpretability**: World model predictions can easily be visualized

Model-based Planning

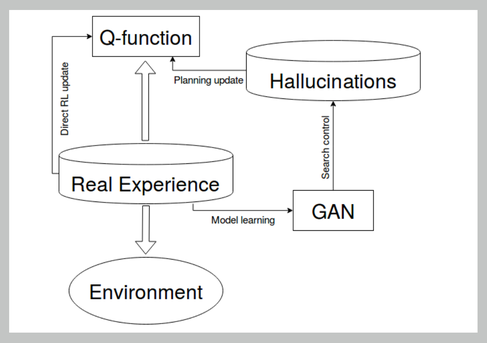


Figure: The general architecture: Real experience, passing back and forth between the environment and the policy affects policy and value functions in much the same way as does simulated experience generated by the model of the environment. Similar to Dyna-Q [3].

The algorithm

Algorithm 1

Initialize $Q(s, a)$ and Model (s, a) for all $s \in \mathcal{S}$ and $a \in A(s)$

loop forever

(a) $S \leftarrow$ current (nonterminal) state

(b) $A \leftarrow \epsilon$-greedy (S, Q)

(c) Take action A; observe resultant reward, R, and state, S'

(d) $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma \max_a Q(S', a) - Q(S, A)]$

(e) GAN$(S, A) \leftarrow R, S'$ (assuming deterministic environment)

loop repeat n times

$S \leftarrow$ random previously observed state

$A \leftarrow$ random action previously taken in S

$R, S' \leftarrow$ GAN(S, A)

$Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma \max_a Q(S', a) - Q(S, A)]$

Learning the Environment

We predict environment output (s', r) from (s, a) using supervised learning.

To do this we adapt conditional GAN losses from [1]

$$L_{\text{GAN}}(G) = \mathbb{E}_y \log D(y|x) + \mathbb{E}_x \log[1 - D(G(x)|x)]$$
$$L_{\text{L1}}(G) = \mathbb{E}_{x,y} ||G(x) - y||_1$$

where x and y are the respective (s, a) and (s', r) tuples.

The objective of our world model now becomes

$$G^* = \arg\min_G \max_D L_{\text{GAN}}(G, D) + \lambda L_{\text{L1}}(G)$$

where λ denotes the relative importance the two losses and $\lambda = 100$ gave the best results.

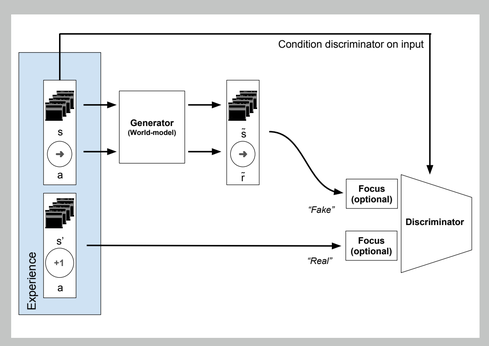
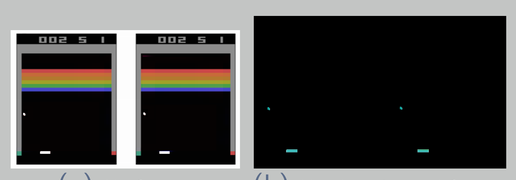


Figure: Generator creates next states and rewards from state-action pairs that fool the discriminator. A (focused) discriminator conditioned on the original state-action pair tries to discriminate between real and generated next states and rewards.

Attention-guided World-Models

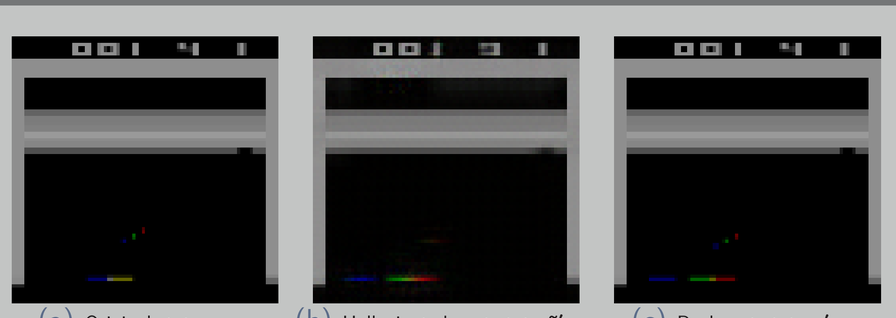
To ensure high-quality generator output in areas that are of importance to the agent, we concatenate a focus map to the inputs of the discriminator, by applying a differentiable focus function $f(\cdot)$ to these inputs. In our case, we use domain-knowledge to come up with a hand-engineered version of function $f(\cdot)$: a mask containing solely pedal and the ball. Potentially, we could replace this with a (self-)attention module that is learned by the agent, such as the one in [4].



(a) No focus (b) Hand-engineered focus

Figure: Possible focus maps

Result: State Mappings



(a) Original state s (b) Hallucinated next state $\tilde{s}'$ (c) Real next state s'

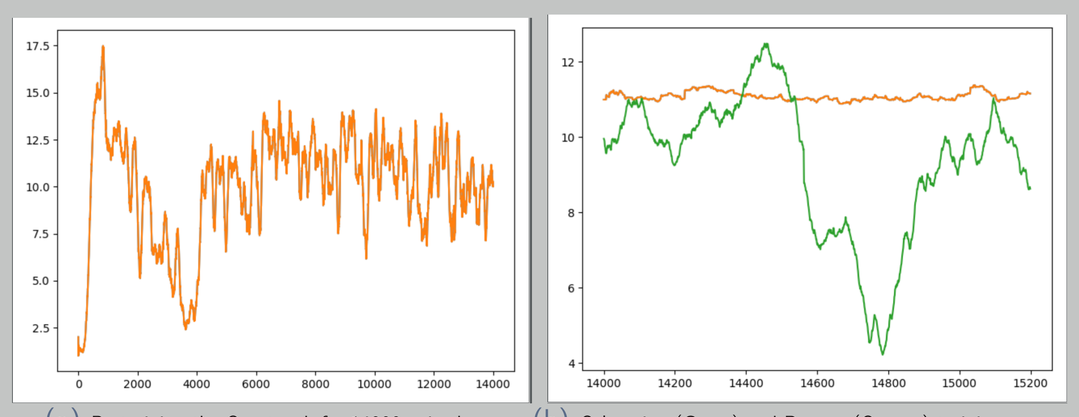
Figure: An example state mapping from GAN with action *left*: We visualize time by plotting greyscaled frames over subsequent time steps (t, $t+1$ and $t+2$) on the respective red, green and blue color-channels. Note from the pedal colors, that the GAN correctly models pedal movement. Also note that there is no ball.

Experimental Design

The task of training a Q network for Atari Breakout is quite challenging and very time consuming. Different hyper parameters were tested on short training sessions that allowed us to find a proper set of hyper-parameters for one big training. We had to adapt this technique due to the limited time of the project. Hyper-paramteters tested: **Discount factor** 0.95, 0.98 and 0.99. **Learning Rate** $\mathbf{10^{-3}}$ and $\mathbf{2 * 10^{-3}}$. **Experience Replay Memory Size** $\mathbf{10^4}$, $\mathbf{10^5}$ and $\mathbf{10^6}$. **Exploration Steps** $\mathbf{10^4}$, $\mathbf{10^5}$ and $\mathbf{10^6}$. The model was trained with q-learning with the best hyper-parameter setting for 14000 episodes, when the training continued both for simple q-learning and Dyna-q. Regarding Dyna-q experiment, the model was trained for 2 hallucinating samples, for every real environment sample.

Results

The cGAN is stabilizing the training procedure but unfortunately the time limitation does not allow us to study thoroughly its effect and confidently evaluate it.



(a) Pretraining the Q-network for 14000 episodes. (b) Q-learning (Green) and Dyna-q (Orange) training behaviours.

Conclusion

In this work,

- we propose a novel model-based RL approach using conditional GANs
- we provide evidence that we can mimic the environment with a GAN
- we found that naively training a GAN does not effectively focus on features that are important to the agent, such as pedal and ball position
- we found that a GAN-based world-model can benefit from more explicit feedback about what is important in the world

Future research

In furture work, we would like to

- assess whether we can replace hand-engineered focus feature with self-attention module that is learned by the Q-network
- extensively tune the model and compare the performance against baselines
- use world model in combination with other planning methods
- model a stochastic environment with stochastic GAN

And if we get even more crazy...

- use causal inference to find important to focus on
- replace GAN with flow-based density approximation
- try to model generator with a recursive function
- model uncertainty of GAN outputs using a bayesian generator

References

1. Isola, Phillip, et al. "Image-to-image translation with conditional adversarial networks." arXiv preprint (2017).
2. Ha, David, and Jrgen Schmidhuber. "World Models." arXiv preprint arXiv:1803.10122 (2018).
3. Sutton, Richard S. "Integrated architectures (...) dynamic programming." Machine Learning Proceedings 1990. 1990. 216-224.
4. Sorokin, Ivan, et al. "Deep attention recurrent Q-network." arXiv preprint arXiv:1512.01693 (2015).
5. Sutton, Richard S., and Andrew G. Barto. Introduction to reinforcement learning. Vol. 135. Cambridge: MIT press, 1998.