World Models

Hao Sun

Mar 2019

Background: RL

- RL considers a discrete-time Markov Decision Process (MDP), defined by (S, A, P, ρ_0, r, γ)

- A stochastic policy π is $S \times A \rightarrow [0,1]$, is to optimize the expected return $\eta(\pi) = E_\tau[\sum_{t=0}^{\infty} \gamma^t \, r(s_t, a_t)]$

Model-Free/ Model-Based

- Model Free (MF): Learning a policy from interactions with the environment directly.
  - Pros:
    - Do not rely on models learned but from samples (learn a Q function)
    - Fast inference
  - Cons:
    - Unstable in optimization
    - Week generalization ability
  - E.g. REINFORCE, DQN, DDPG, A3C …

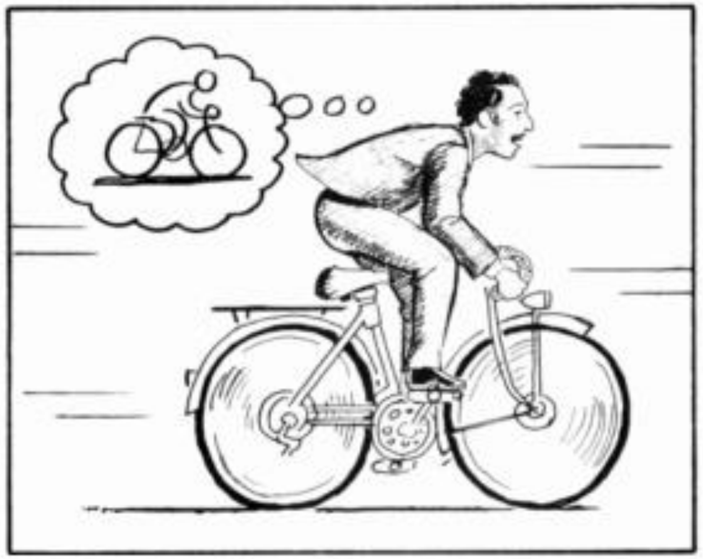
Model-Free/ Model-Based

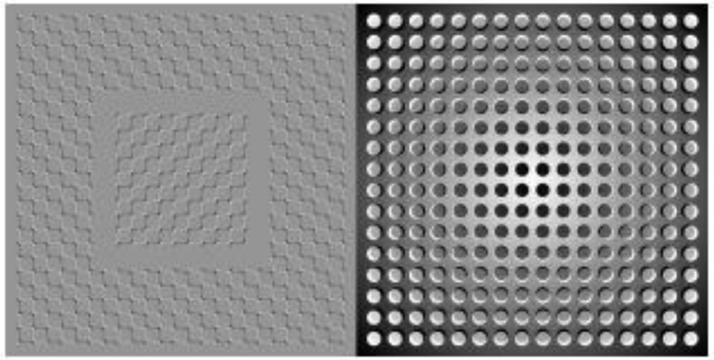
- Model Based (MB): Learning a transition model, or P mentioned above, so that one may use such transition model to optimize a policy.
  - Pros:
    - Learning Q value is hard, but learning transition model is relatively easy (SL)
    - Fast transfer into new environment
  - Cons:
    - Error accumulation
  - E.g. World Models

World Models, David Ha, 2018

- *The image of the world around us, which we carry in our head, is just a model. Nobody in his head imagines all the world, government or country. He has only selected concepts, and relationships between them, and uses those to represent the real system.*

-Jay Wright Forrester,
the father of system dynamics





A bottleneck of MF

- Many MF methods use small neural networks
- Bottlenecked by the credit assignment problem
- It is hard to learn millions of weights of large models

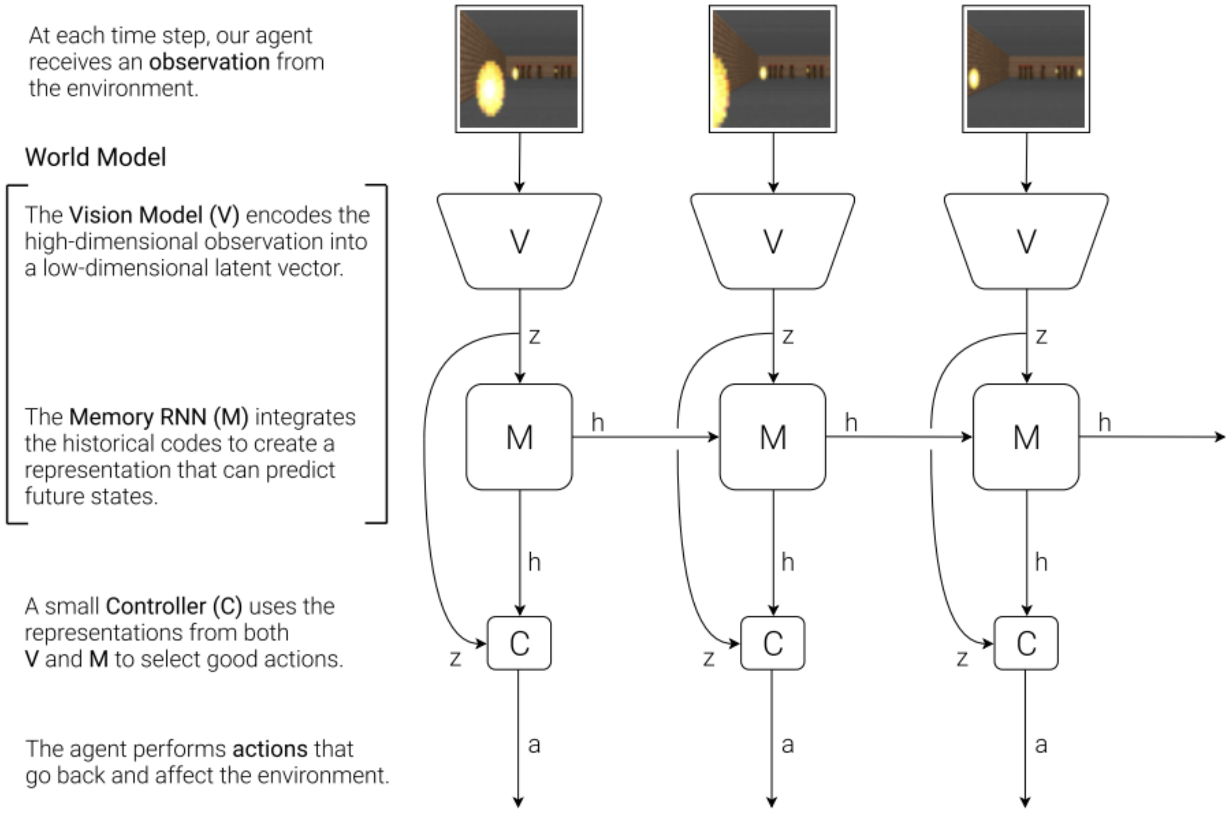
- Details:
  - MF methods optimize a q-function or q-value estimator, and leverage the trajectory samples collected by running a policy to revise the estimator. (Temporal Difference, TD-methods)
  - The process is unstable, especially in case of sparse reward problems.

Solution of World Models

- Back Propagation methods can be used to train large NN based agents efficiently.
- Credit assignment?
  - Split the problem into two parts:
    - Learning a world model using VAE and RNN, with large NN
    - Optimize in the world model using evolution algorithms, with small NN
  - Model based method

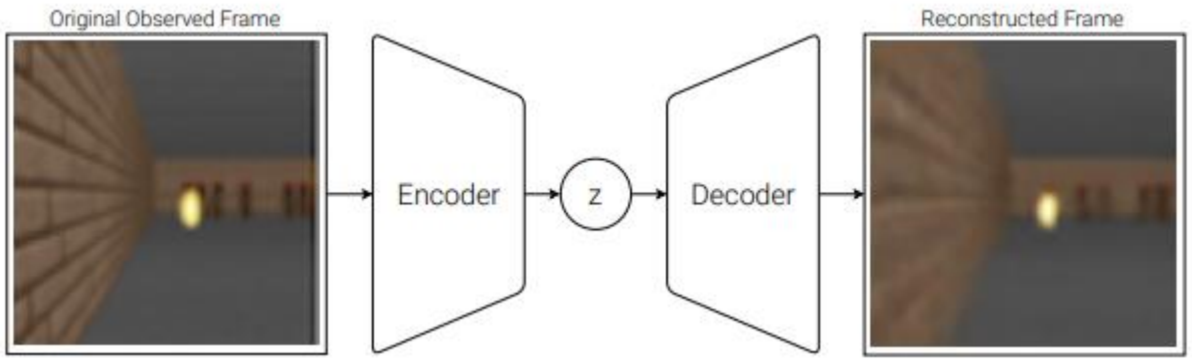
The world Model

- Three parts
  - Vision Model
  - Memory Model
  - Controller

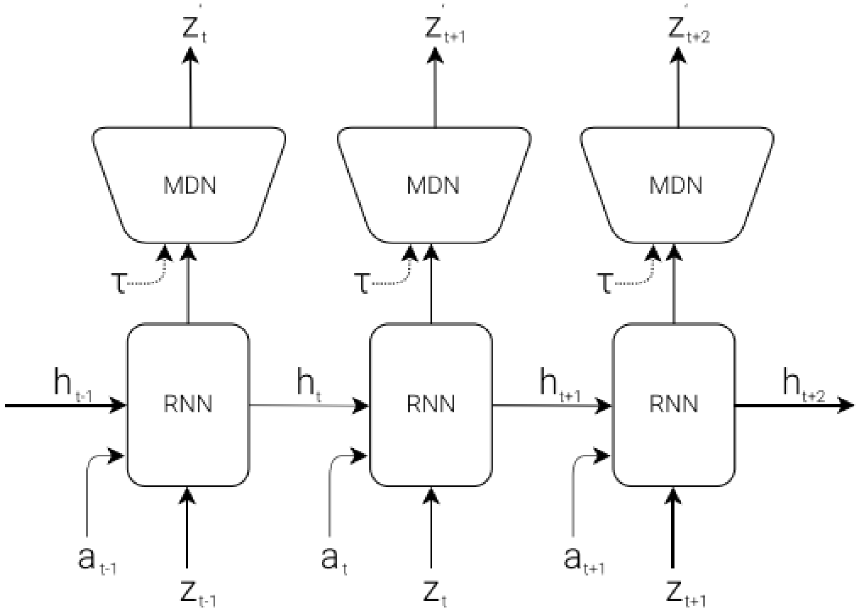


Three parts

- Vision Module:
  VAE



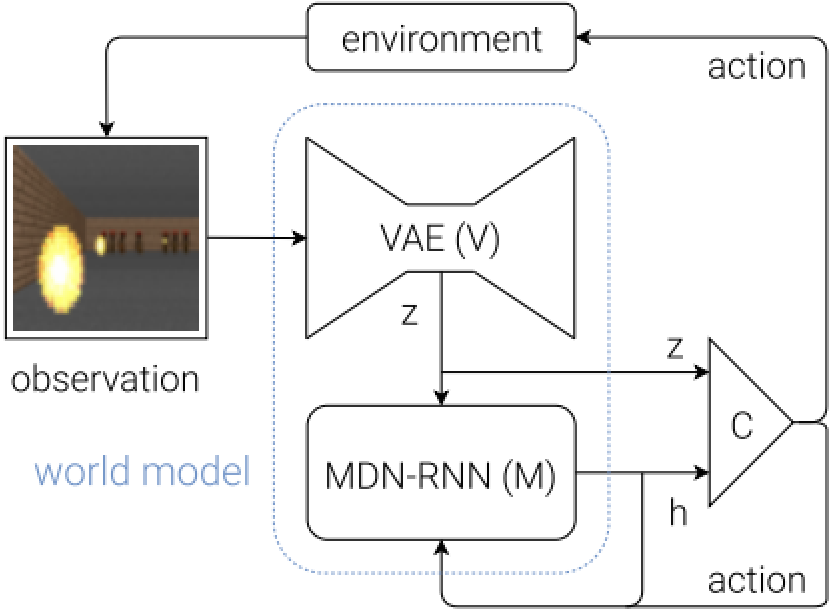
- Memory Module:
  RNN (with MDN)



- Controller:
  $a_t = W_c[z_t, h_t] + b_c$

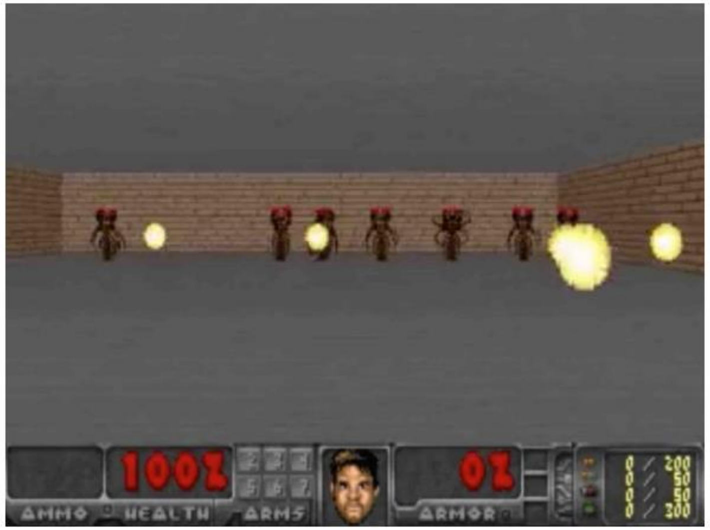
Putting them together

- observation-> z
- z, a -> h
- [z, h] -> a
- a -> observation



Experiments

- CarRacing-v0, gym
  - Collect 10,000 rollouts from a random policy.
  - Train VAE (V) to encode frames into $z \in R^{32}$.
  - Train MDN-RNN (M) to model $P(z_{t+1}|a_t, z_t, h_t)$.
  - Define Controller (C) as $a_t = W_c[z_t, h_t] + b_c$.
  - Use CMA-ES to solve for a W_c and b_c that maximizes the expected cumulative reward.
- VizDoom



Results

- CarRacing

METHOD	AVG. SCORE
DQN (PRIEUR, 2017)	343 ± 18
A3C (CONTINUOUS) (JANG ET AL., 2017)	591 ± 45
A3C (DISCRETE) (KHAN & ELIBOL, 2016)	652 ± 10
CEOBILLIONAIRE (GYM LEADERBOARD)	838 ± 11
V MODEL	632 ± 251
V MODEL WITH HIDDEN LAYER	788 ± 141
FULL WORLD MODEL	$\mathbf{906 \pm 21}$

Table 1. `CarRacing-v0` scores achieved using various methods.

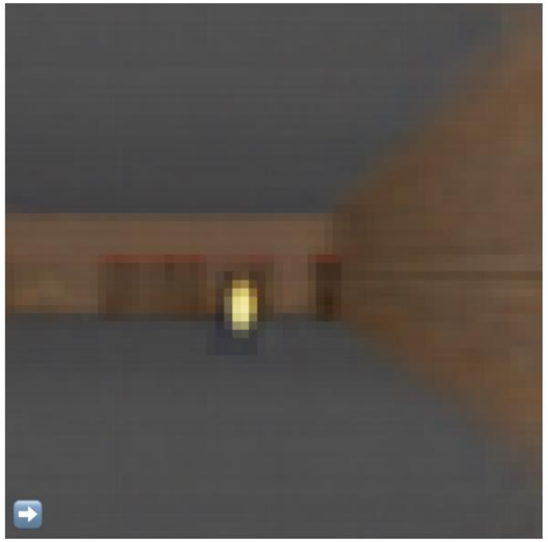
- VizDoom

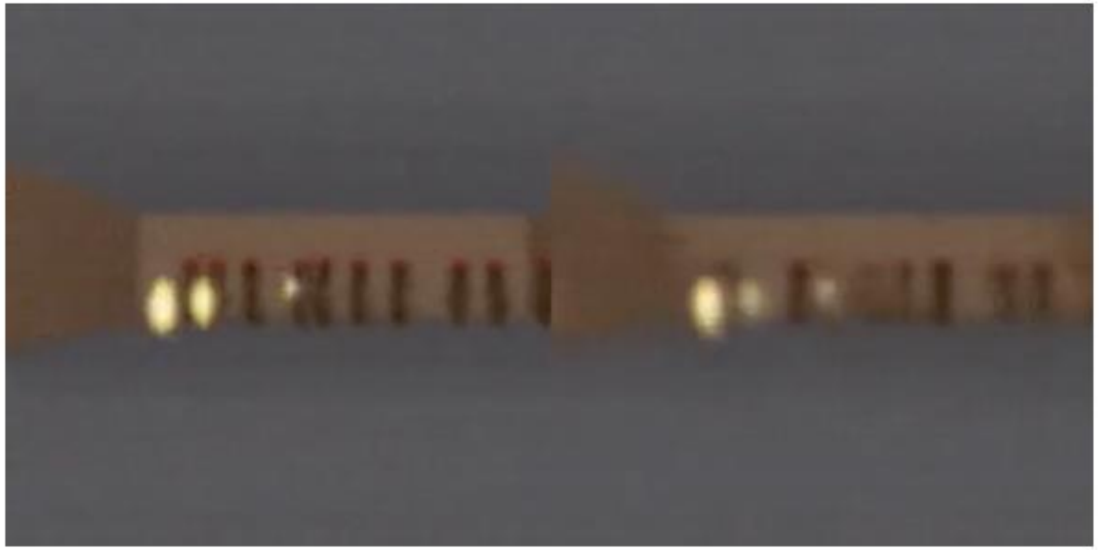
TEMPERATURE τ	VIRTUAL SCORE	ACTUAL SCORE
0.10	2086 ± 140	193 ± 58
0.50	2060 ± 277	196 ± 50
1.00	1145 ± 690	868 ± 511
1.15	918 ± 546	1092 ± 556
1.30	732 ± 269	753 ± 139
RANDOM POLICY	N/A	210 ± 108
GYM LEADER	N/A	820 ± 58

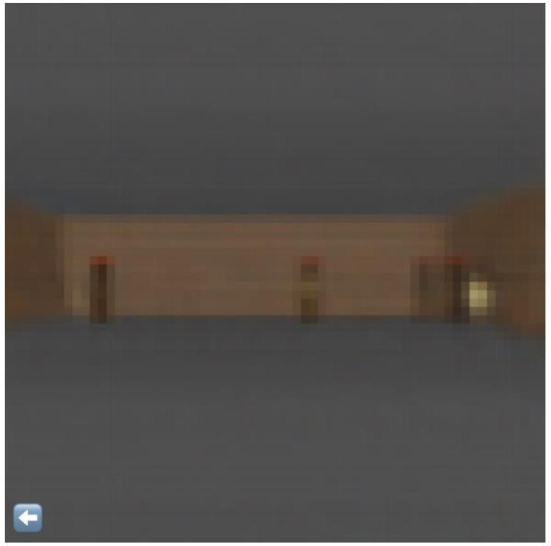
Table 2. Take Cover scores at various temperature settings.

Cheating behaviors

- Training inside of the dream
- ‘Bugs’ of the environment
- Iteratively training







References & Extensions

- Ha D , Schmidhuber, Jürgen. World Models[J]. 2018.
- Hafner D , Lillicrap T , Fischer I , et al. Learning Latent Dynamics for Planning from Pixels[J]. 2018.
- https://worldmodels.github.io/
- https://planetrl.github.io/