Generative World Models:
The Dreamer Models

Sanjiban Choudhury
The story so far ...

Robots have to act in the world

Hence, we learned various algorithms for decision making

But we assumed that we can observe the "state"
The story so far …

But in the real world, no one tells you the “state”

All you see are observations

How do we learn from observations?
The story so far ...

Our focus in this and future lectures will turn to learning representations
Models.
What is a model?
What is a model?

\[ S_t, a_t \]

\[ \downarrow \]

\[ \text{Model} \]

\[ \downarrow \]

\[ S_{t+1} \]
What is a model?

\[ P_\theta(s_{t+1} \mid s_t, a_t) \]
Learning Models
Models: From Simple to Complex
Models: From Simple to Complex

- Physics Models
- Simple
- Known state
- Strong prior on dynamics
Models: From Simple to Complex

Physics Models

Simple

Known state

Strong prior on dynamics

Motion Models

Known state

Unknown dynamics
Models: From Simple to Complex

Physics Models
- Known state
- Strong prior on dynamics

Motion Models
- Known state
- Unknown dynamics

Open World Models
- Unknown state
- Unknown dynamics
Activity!
Modelling Tamago Sushi
Think-Pair-Share!

Think (30 sec): How would you model making tamago sushi?

Pair: Find a partner

Share (45 sec): Partners exchange ideas
Challenges with learning complex models

Challenge 1: Can’t see state, only get high-dimensional observations

Challenge 2: Planning with complex dynamics
How can we learn latent low-dimensional state from high-dimensional observations?
Idea: Use “auto-encoder” trick from computer vision
Reconstructed image

\[ \hat{X} = \mathcal{F}(X) \]

From MIT 6.8300/6.8301: Advances in Computer Vision
Action "Flip"
Previous State $S_{t-1}$

State

Action "Flip"

$S_t$
The DREAMER Algorithms
Mastering Diverse Domains through World Models

Danijar Hafner,1,2 Jurgis Pasukonis,1 Jimmy Ba,2 Timothy Lillicrap1

1DeepMind  2University of Toronto
MineRL Diamond Challenge
MineRL Diamond Challenge

1. Gather Wood
2. Create Wood Pickaxe
3. Mine Stone and Create Stone Pickaxe
4. Mine Iron Ore
5. Create Furnace
6. Smelt Iron and Create Iron Pickaxe
7. Search
8. Mine Diamond
DreamerV3 solved this task!
The DREAMER Algorithm
DREAM TO CONTROL: LEARNING BEHAVIORS BY LATENT IMAGINATION

Danijar Hafner *  
University of Toronto  
Google Brain

Timothy Lillicrap  
DeepMind

Jimmy Ba  
University of Toronto

Mohammad Norouzi  
Google Brain

2020
Look at the videos below

Is this from the actual simulator or predictions made by a model?
Look at the videos below

Boxing  Freeway  Frostbite  Collect Objects  Watermaze

Sparse Cartpole  Acrobat Swingup  Hopper Hop  Walker Run  Quadruped Run

Predictions by a model!
Recap: Model-based RL

(Ross & Bagnell, 2012)
How does DREAMER fit a model?
Goal: Fit a Model given data

Given Data:
Observations, rewards, actions
Goal: Fit a Model given data

Given:
Observations, rewards, actions

Predict:
States, Dynamics Function, Reward Function
$p_\theta(s_t | o_t, s_{t-1}, a_{t-1})$

State Encoder

encode images
compute states
\[ \ell = (r_t - \hat{r}_t)^2 \]

\[ q_\theta(r_t | s_t) \]

Reward Decoder
\[ \ell = (o_t - \hat{o}_t)^2 \]

\[ q_\theta(o_t \mid s_t) \]

Observation Decoder
\[ q_\theta(s_t | s_{t-1}, a_{t-1}) \]

Dynamics Function
Results: Learning World Model
Results: Learning World Model
How does DREAMER do planning?

1. Collect Expert Data
2. Fit Model
3. Planner
   - Rollout Policy
Goal: Learn a Policy using Actor-Critic

\[
\pi_{\phi}(a_t \mid s_t) \quad V_{\psi}(s_t)
\]

Actor \quad Critic

From rollouts in the model

\[
q_\theta(s_t \mid s_{t-1}, a_{t-1})
\]
Recall: Actor-Critic

Start with an arbitrary initial policy $\pi_\theta(a \mid s)$

while not converged do

Roll-out $\pi_\phi(a \mid s)$ in the model $q_\theta(s' \mid s, a)$ to collect trajectories $D = \{s^i, a^i, r^i, s_{i+1}^i\}_{i=1}^N$

Fit value function $V_\psi(s^i)$ using TD, i.e. minimize $(r^i + \gamma V_\psi(s_{i+1}^i) - V_\psi(s^i))^2$

Compute advantage $\hat{A}(s^i, a^i) = r(s^i, a^i) + \gamma V_\psi(s_{i+1}^i) - V_\psi(s^i)$

Compute gradient

$$\nabla_\phi J(\phi) = \frac{1}{N} \left[ \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\phi(a_t^i \mid s_t^i) \hat{A}(s_t^i, a_t^i) \right]$$

Update parameters $\phi \leftarrow \phi + \alpha \nabla_\phi J(\phi)$
encode images
Rollout policy

$\pi_\phi(a_t | s_t)$

encode images
imagine ahead
Predict rewards (Freeze gradients)

$q_{\theta}(r_t | s_t)$
 encode images
imagine ahead
predict rewards
predict values

Update critic

$V_{\psi}(s_t)$
encode images
imagine ahead
predict rewards
predict values

Update actor
$\pi_\phi(a_t | s_t)$
DREAMER: Results
DREAMER is a template for Model-based RL

But there are many challenges as we scale to harder real-world applications
DREAMER V2:

Tackling the world of Atari Games
MASTERING ATARI WITH DISCRETE WORLD MODELS

Danijar Hafner
Google Research
Timothy Lillicrap
DeepMind
Mohammad Norouzi
Google Research
Jimmy Ba
University of Toronto
Atari was hard for Model Based RL
DreamerV2 beats all model free!

Atari Performance

- **Model-based**
- **Model-free**

Human Gamer
Montezuma’s Revenge:

A really challenging Atari Game!
Challenge: Dreamer V1 predicts a single mode of dynamics
Dreamer V1 predicts single mode dynamics
Idea: Predict multiple discrete modes!