Prediction-based Exploration

Christopher Mutschler

Agenda

- Motivation, Problem Definition & Multi-Armed Bandits
- Classic Exploration Strategies
  - Epsilon Greedy
  - (Bayesian) Upper Confidence Bounds
  - Thomson Sampling
- **Exploration in Deep RL**
  - Count-based Exploration: Density Models, Hashing
  - **Prediction-based Exploration:**
    - Forward Dynamics
    - Random Networks
    - Physical Properties
  - Memory-based Exploration:
    - Episodic Memory
    - Direct Exploration
- Summary and Outlook

Prediction-based Exploration

- So far, we derived the bonus with respect to the *novelty* of states we encounter
  - The bonus correlates with the state visitation
  - We encourage the agent to look for states he did not see that often
- However, we also could interpret intrinsic motivation more widely in terms of *curiosity*
  - Obtaining knowledge about the environment
  - Familiarity with the environment dynamics, reward structure, …
- In RL, the idea of using a prediction model actually dates back to 1991[1]

https://www.youtube.com/watch?v=8vNxjwt2AqY

A Possibility for Implementing Curiosity and Boredom in Model-Building Neural Controllers

Jürgen Schmidhuber*
TUM

In J. A. Meyer and S. W. Wilson, editors, Proc. of the International Conference on Simulation of Adaptive Behavior: From Animals to Animats, pages 222-227. MIT Press/Bradford Books, 1991.

Abstract

This paper introduces a framework for 'curious neural controllers' which employ an adaptive world model for goal directed on-line learning.

First an on-line reinforcement learning algorithm for autonomous 'animats' is described. The algorithm is based on two fully recurrent 'self-supervised' continually running networks which learn in parallel. One of the networks learns to represent a complete model of the environmental dynamics and is called the 'model network'. It provides complete 'credit assignment paths' into the past for the second network which controls the animats physical actions in a possibly reactive environment. The animats goal is to maximize cumulative reinforcement and minimize cumulative 'pain'.

The algorithm has properties which allow to implement something like *the desire to improve the model network's knowledge about the world.* This is related to *curiosity.* It is described how the particular algorithm (as well as similar model-building algorithms) may be augmented by dynamic *curiosity* and *boredom* in a natural manner. This may be done by introducing (delayed) reinforcement for actions that increase the model network's knowledge about the world. This in turn requires the model network *to model its own ignorance*, thus showing a rudimentary form of *self-introspective* behavior.

[1] *Jürgen Schmidhuber: A Possibility for Implementing Curiosity and Boredom in Model-Builing Neural Controllers. Intl. Conf. Simulation of Adaptive Behavior. 1991.*

Predicting Models: Forward Dynamics

- Idea of the **forward dynamics prediction model**:
  - The agent learns a parameterized function f_θ such that:

$$f_\theta : (s_t, a_t) \rightarrow s_{t+1}$$

  - Derive a reward bonus based on the prediction error of the dynamics model

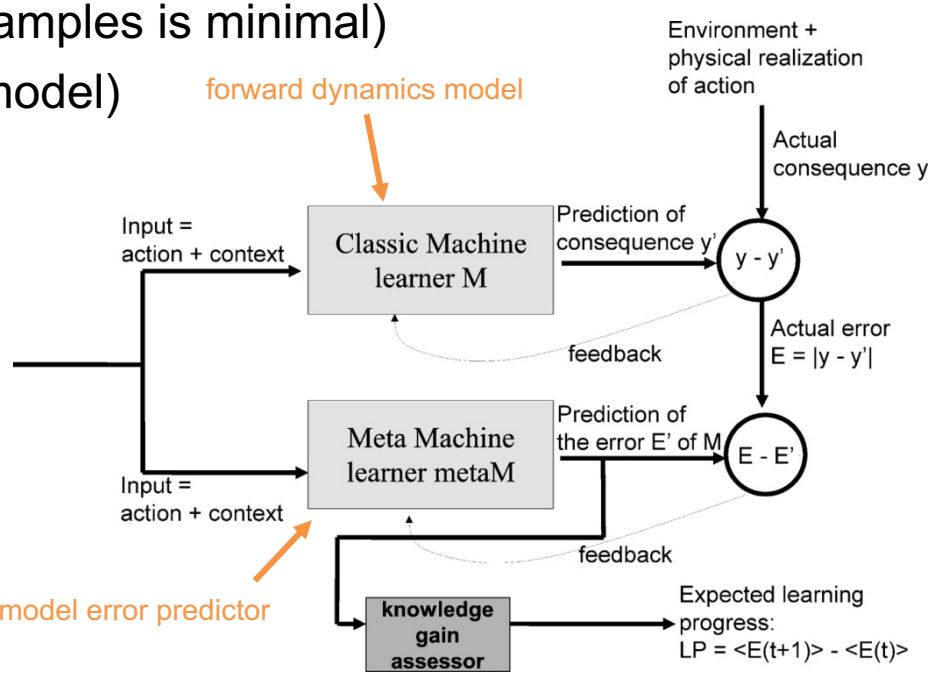
$$e(s_t, a_t) = \| f(s_t, a_t) - s_{t+1} \|_2^2$$

    - Large prediction error: high bonus (as we encountered something unusual/unknown)
    - Low prediction error: low bonus (as we have seen this coming)
  - Our agent uses all the experience samples (s_t, a_t, s_{t+1}) collected so far and retrains its prediction model as it interacts with the environment

Predicting Models: Forward Dynamics

Intelligent Adaptive Curiosity (IAC)[1]

- A memory M stores all the experiences encounted so far: $M = \{(s_t, a_t, s_{t+1})\}$
- Idea:
  - IAC (incrementally) splits the sensorimotor space into *regions* $\mathcal{R}_{0\ldots n}$ (criterion: the sum of variances of the two sets weighted by #samples is minimal)
  - Each region $\mathcal{R}_n$ has an associated *expert* (forward dynamics model) that is trained using the data from its region
- Learning:
  - With each new action the prediction error of the forward dynamics model is calculated using the MSE and put in a sliding window associated to that region
  - The decrease in the mean error rate is given as a reward to the agent



[1] *Pierre-Yves Oudeyer et al.: Intrinsic Motivation Systems for Autonomous Mental Development. Trans. Evol. Computation. 11(2). 2007.*

Predicting Forward Dynamics

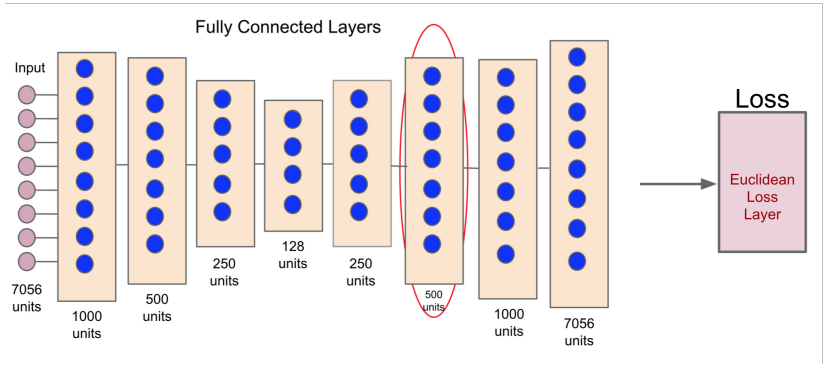
Deep Predictive Models[1]

- Predicting high-dimensional state spaces (images) can become very difficult
- Train a forward dynamics model in an encoding space ϕ (train an autoencoder):

$$f_\phi : (\phi(s_t), a_t) \rightarrow \phi(s_{t+1})$$

- Normalize the prediction error at time T by the maximum error so far:

$$\bar{e}_t = \frac{e_t}{\max\limits_{i \leq t} e_i}$$



- Define the extrinsic reward accordingly (C is a decay parameter):

$$r_t^i = \left(\frac{e_t(s_t, a_t)}{t \cdot C} \right)$$

- The autoencoder can be trained upfront using images collected randomly or trained along with the policy and being updated steadily.

[1] *Stadie, Levine, Abbeel: Incentivizing Exploration in Reinforcement Learning with Deep Predictive Models. 2015.*

Predicting Forward Dynamics

Intrinsic Curiosity Module (ICM)[1]

- Instead of an autoencoder ICM trains the state space encoding $\phi(s_t)$ with a self-supervised *inverse dynamics* model

learn to explore in Level-1 explore faster in Level-2

- Motivation:
  - Predicting s_{t+1} given (s_t, a_t) is not always easy as many factors in the environment cannot be controlled/affected by the agent
  - Popular example: imagine this tree with leaves
  - Such factors should not be part of the encoded state space as the agent should not base its decision based on these factors

- Solution: Learn an inverse dynamics model g:

$$g: \left(\phi(s_t), \phi(s_{t+1})\right) \rightarrow a_t$$

- The feature space then only captures those changes in the environment related to actions that the agent takes, and ignores the rest

[1] *Deepak Pathak et al.: Curiosity-driven Exploration by Self-Supervised Prediction. ICML 2017.*

Predicting Forward Dynamics

Intrinsic Curiosity Module (ICM)[1], given

- a forward model f with parameters θ_F
- an inverse dynamics model g with parameters θ_I
- and an observation (s_t, a_t, s_{t+1})

$$\hat{a}_t = g(\phi(s_t), \phi(s_{t+1}); \theta_I)$$

$$\hat{\phi}(s_{t+1}) = f(\phi(s_t), a_t; \theta_F)$$

$$r_t^i = \frac{\eta}{2}\left\|\hat{\phi}(s_{t+1}) - \phi(s_{t+1})\right\|_2^2$$

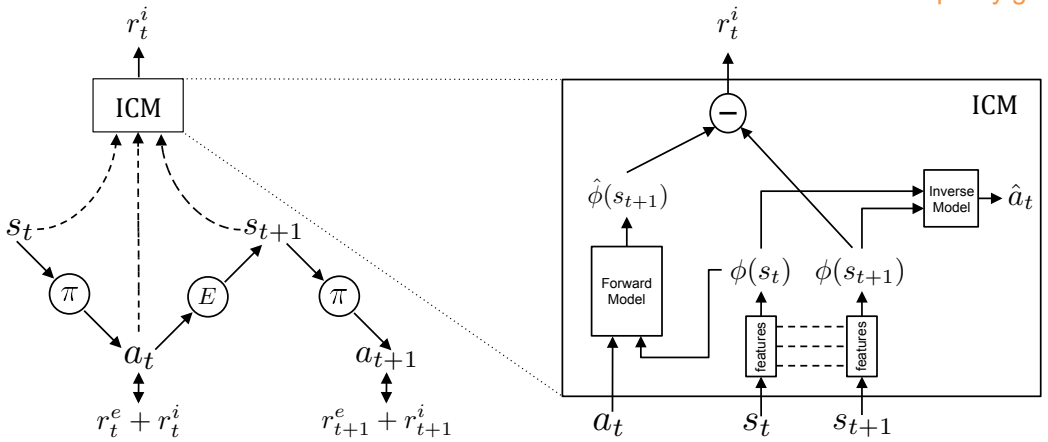
- The policy is jointly optimized as a whole:

if actions are discrete: softmax ML under multinomial distribution

$$\min_{\theta_P, \theta_I, \theta_F}\left[-\lambda \mathbb{E}_{\pi(s_t;\theta_P)}\left[\sum_t r_t\right] + (1-\beta)L_I + \beta L_F\right]$$

policy gradient loss

$$L_F\left(\phi(s_t), \hat{\phi}(s_{t+1})\right) = \frac{1}{2}\left\|\hat{\phi}(s_{t+1}) - \phi(s_{t+1})\right\|_2^2$$



[1] *Deepak Pathak et al.: Curiosity-driven Exploration by Self-Supervised Prediction. ICML 2017.*

Predicting Forward Dynamics

Intrinsic Curiosity Module (ICM)[1]: Large-Scale Study

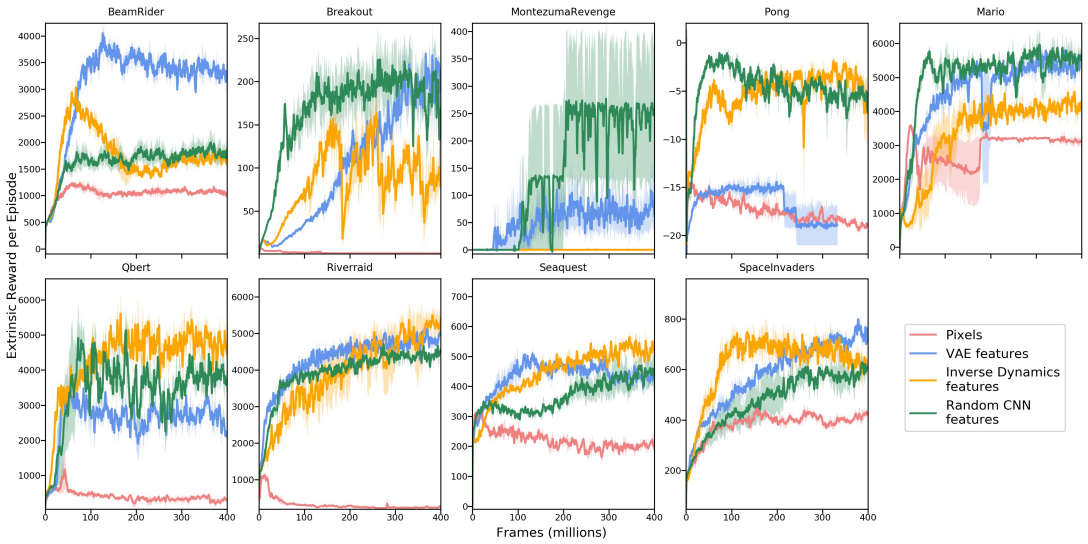
- Analyze the influence of curiosity by purely using the intrinsic reward:

$$r_t = r_t^i = \|f(s_t, a_t) - \phi(s_{t+1})\|_2^2$$

- What requirements must ϕ satisfy? When does it work best?

compact, sufficient, stable

	VAE	IDF	RF	Pixels
stable	✗	✗	✓	?
compact	✓	✓	?	✗
sufficient	✓	?	?	✓



→ Random CNN features are simple yet surprisingly strong!
→ However, IDF generalizes better (transfer knowledge from Super Mario Bros. Level 1 → Level 2)

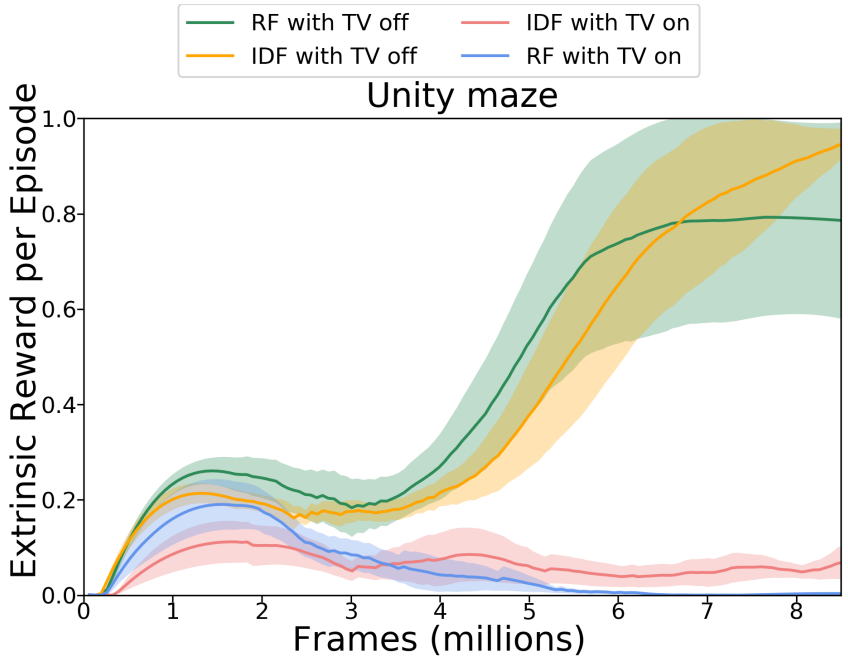
Details on experiments:
- Robust learning: PPO
- Reward, Advantage, Observation Normalization
- 128 parallel actors
- Feature normalization
- Infinite horizon (avoid done flag)

[1] *Yuri Burda et al.: Large-Scale Study of Curiosity-Driven Learning. ICLR 2019.*

Predicting Forward Dynamics

Intrinsic Curiosity Module (ICM)[1]: Noisy TV

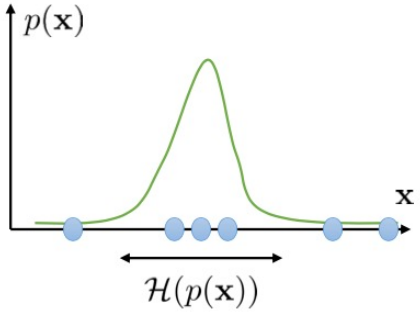
- Noisy TV drastically slows down the learning as extrinsic rewards are considerably lower in time
  → stochasticity of the environment poses problems
- Stochasticity is not always a problem, sometimes the agent escapes



[1] *Yuri Burda et al.: Large-Scale Study of Curiosity-Driven Learning. ICLR 2019.*

Predicting Forward Dynamics

Variational Information Maximizing Exploration (VIME)[1]:



[2]

- Some useful theoretic quantities first[2]:
  - $p(x)$: distribution (e.g., over observations x)
  - $\mathcal{H}\big(p(x)\big) = -\mathbb{E}_{x\sim p(x)}[\log p(x)]$: entropy (how broad is $p(x)$)
  - $\pi(s)$: state marginal distribution of policy π
  - $\mathcal{H}(\pi(s))$: state marginal entropy of policy π (quantifies *coverage*)
  - $\mathcal{I}(s_{t+1}; a_t) = \mathcal{H}(s_{t+1}) - \mathcal{H}(s_{t+1}|a_t)$: mutual information (empowerment)
- Idea of VINE:
  - Given: environment transition function $\mathcal{P}$ and forward prediction model $p(s_{t+1}|s_t, a_t; \theta), \theta \in \Theta$
  - We see a trajectory $\xi_t = \{s_1, a_1, \dots s_t\}$
  - We want to reduce entropy whenever we acquire new knowledge (see new states), i.e., maximize:

$$\sum_t H(\Theta|\xi_t, a_t) - H(\Theta|s_{t+1}, \xi_t, a_t) = \cdots = \mathbb{E}_{s_{t+1}\sim\mathcal{P}(\cdot|\xi_t,a_t)}[D_{KL}(p(\theta|\xi_t, a_t, s_{t+1})\|p(\theta|\xi_t))]$$

[2] *See also Sergey Levine CS285, Lecture 14*

[1] *Rein Houthooft et al.: VIME: Variational Information Maximization Exploration. NIPS 2016.*

Predicting Forward Dynamics

Variational Information Maximizing Exploration (VIME)[1]:

- Unfortunately, computing the posterior $p(\theta|\xi_t, a_t, s_{t+1})$ is intractable:

$$p(\theta|\xi_t, a_t, s_{t+1}) = \frac{p(\theta|\xi_t, a_t)p(s_{t+1}|\xi_t, a_t; \theta)}{p(s_{t+1}|\xi_t, a_t)}$$

$$= \frac{p(\theta|\xi_t)p(s_{t+1}|\xi_t, a_t; \theta)}{p(s_{t+1}|\xi_t, a_t)}$$

action does not affect the belief

$$= \frac{p(\theta|\xi_t)p(s_{t+1}|\xi_t, a_t; \theta)}{\int_\Theta p(s_{t+1}|\xi_t, a_t; \theta)p(\theta|\xi_t)d\theta}$$

hard to compute directly

- As it is difficult to compute $p(\theta|\xi_t)$, we approximate it with an alternative distribution $q_\phi(\theta)$
- Variational Inference: using the variational lower bound: maximizing $q_\phi(\theta)$ is equivalent to
  - Maximizing $p(\xi_t|\theta)$ and minimizing $D_{KL}\left[q_{\phi_{t+1}}(\theta)\|p(\theta)\right]$

[1] *Rein Houthooft et al.: VIME: Variational Information Maximization Exploration. NIPS 2016.*

Predicting Forward Dynamics

Variational Information Maximizing Exploration (VIME)[1]:

- Using the approximated distribution q, the intrinsic reward is

$$r_t^i = D_{KL}\left[q_{\phi_{t+1}}(\theta)||q_{\phi_t}(\theta)\right],$$

  where ϕ_{t+1} are the parameters of q after seeing a_t and s_{t+1}

- Normalize by division by moving median to KL-divs when using as a reward

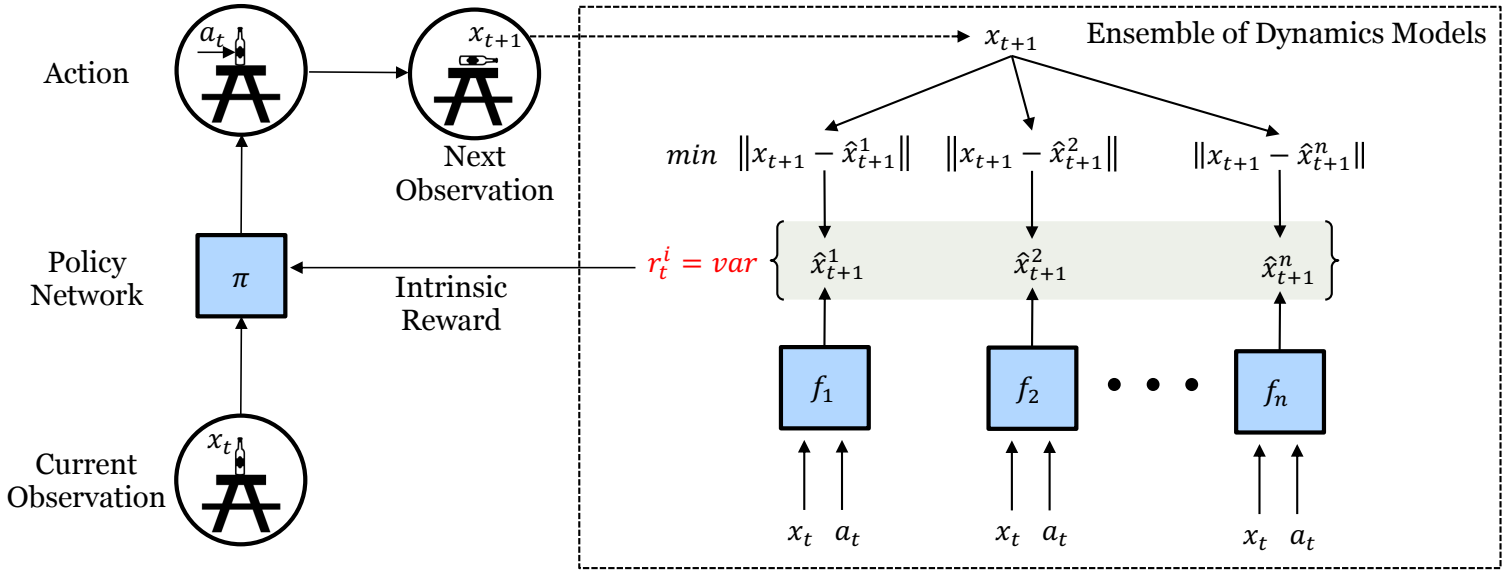
- VIME uses a Bayesian neural network (BNN) to maintain a distribution over weights
  - The weights are modeled as Gaussians, and we can sample $\theta \sim q_\phi(\cdot)$
  - KL-Divergence is estimated from the 2nd-order Taylor-expansion using the FIM (which is easy to compute as the Gaussians result in a diagonal covariance matrix)

[1] *Rein Houthooft et al.: VIME: Variational Information Maximization Exploration. NIPS 2016.*

Predicting Forward Dynamics

Self-Supervised Exploration via Disagreement[1]:

- Use an ensemble of prediction models and use their disagreement as bonus
- High disagreement → low confidence → needs more exploration
- r_t^i is differentiable → intrinsic reward can be directly optimized
- very efficient differentiable approach



[1] Deepal Pathak et al.: Self-Supervised Exploration via Disagreement. ICML 2019.

Prediction Models: Random Networks

- From the noisy TV we can identify sources of prediction errors[1]

But this is our basic assumption – we need this

- The prediction error is high
  1. … where the predictor fails to generalize from previously seen examples.
     Novel experience then corresponds to high prediction error
  2. … because the prediction target is stochastic
  3. … because information necessary for the prediction is missing, or
     the model class of predictors is too limited to fit the complexity of the target function.

- How can we avoid the issues raised by (2) and (3)?

- What if the focus of our prediction task is not on environment dynamics at all?

- **Don’t care about the dynamics? Sounds crazy? Just wait…**

[1] *Yuri Burda et al.: Exploration by Random Network Distillation. ICLR 2019.*

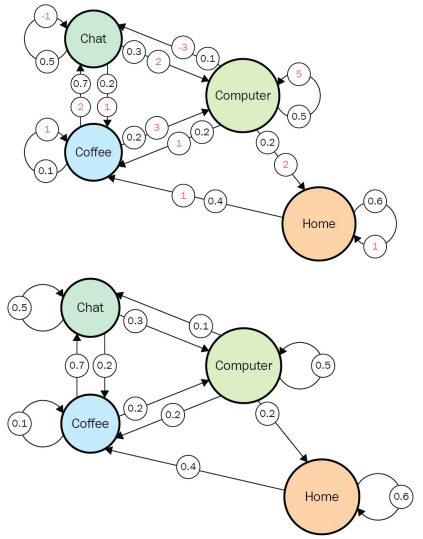
Prediction Models: Random Networks

Directed Outreaching Reinforcement Action-Selection (DORA)[1]

- We use MDPs:
  - (1) the original MDP
  - (2) a copy of (1) with $r(s,a) = 0, \forall s \in \mathcal{S}, a \in \mathcal{A}$
- We learn Q on (2), and we call the entries "E"-values
  - So, we learn $E(s,a)$
  - "E"-values are initialized with 1
  - The model should predict all the E-values to be 0
- State-action pairs with *high* E-values lack information
- This is very similar to *visit counters*
- Given the predicted E-value $E(s_t, a_t)$, the exploration bonus is

$$r_i = \frac{1}{\sqrt{-\log E(s_t, a_t)}}$$

"If there's a place you gotta go - I'm the one you need to know."
(Map, Dora The Explorer)



[1] *Leshem Choshen et al.: DORA The Explorer: Directed Outreaching Reinforcement Action-Selection. ICLR 2018.*

Prediction Models: Random Networks

Directed Outreaching Reinforcement Action-Selection (DORA)[1]

Input: Stochastic action-selection rule f, learning rate α, Exploration discount factor γ_E
initialize $Q(s,a) = 0$, $E(s,a) = 1$;
foreach *episode* **do**
  init s;
  while *not terminated* **do**
    Choose $a = \arg\max_x \log f_Q(x|s) - \log\log_{1-\alpha} E(s,x)$;
    Observe transitions (s, a, r, s', a');
    $Q(s,a) \leftarrow (1-\alpha)\, Q(s,a) + \alpha\,(r + \gamma \max_x Q(s',x))$;
    $E(s,a) \leftarrow (1-\alpha)\, E(s,a) + \alpha\gamma_E E(s',a')$;
  end
end

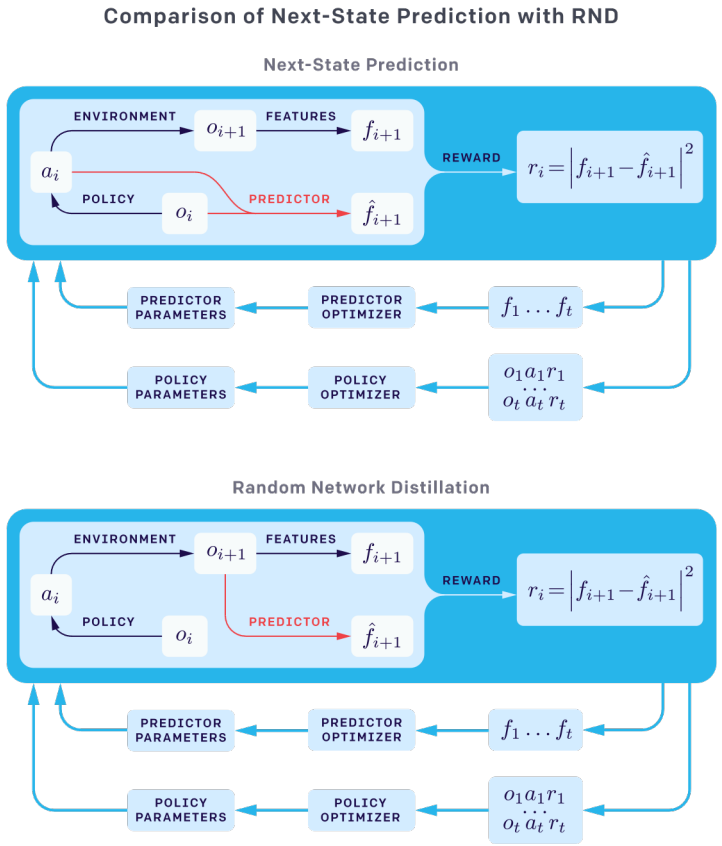
[1] *Leshem Choshen et al.: DORA The Explorer: Directed Outreaching Reinforcement Action-Selection. ICLR 2018.*

Prediction Models: Random Networks

Random Network Distillation (RND)[1,2]

- Similar idea: predict something that is independent from the main task
- We use two neural networks:
  1. A randomly initialized but **fixed** neural network to transform a state into a feature space: $f(s_t)$
  2. A network $\hat{f}(s_t;\theta)$ that we train to predict the same features as the fixed network

  → We want $\hat{f}(s_t;\theta) = f(s_t)$
- Intuition: Similar states have similar features
  - And if we have already seen them, we should also have a lower error on predicting them!
- We use an exploration bonus: $r^i(s_t) = \left\|\hat{f}(s_t;\theta) - f(s_t)\right\|_2^2$



[1] *Yuri Burda et al.: Exploration by Random Network Distillation. ICLR 2019.*
[2] *https://openai.com/blog/reinforcement-learning-with-prediction-based-rewards/*

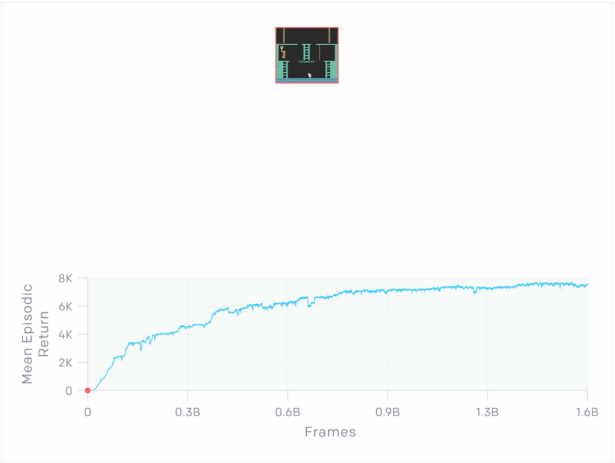
Random Network Distillation

Random Network Distillation (RND)

- Advantage of synthetic prediction problem:
  - The fixed network makes the prediction target deterministic (bypassing issue #2)
  - It is inside the class of functions that the predictor can represent (bypassing issue #3) if the predictor and the target network have the same architecture.

- Results:
  - RND works well for hard-exploration problems
    → maximizing RND bonus finds half of the rooms in Montezuma's Revenge
  - Normalization is important! The scale of the rewards is tricky to adjust given a random network as prediction target
    →Normalize by a running estimate of standard deviations of intrinsic return

  - Non-episodic settings work better, especially in cases without extrinsic rewards (the return is not truncated at *game over* and intrinsic return can spread across multiple episodes)



[1] *https://openai.com/blog/reinforcement-learning-with-prediction-based-rewards/*

RND problem in episodic tasks[1]

2.3 Combining intrinsic and extrinsic returns

In preliminary experiments that used only intrinsic rewards, treating the problem as non-episodic resulted in better exploration. In that setting the return is not truncated at "game over". We argue that this is a natural way to do exploration in simulated environments, since the agent's intrinsic return should be related to all the novel states that it could find in the future, regardless of whether they all occur in one episode or are spread over several. It is also argued in (Burda et al., 2018) that using episodic intrinsic rewards can leak information about the task to the agent.

We also argue that this is closer to how humans explore games. For example let's say Alice is playing a videogame and is attempting a tricky maneuver to reach a suspected secret room. Because the maneuver is tricky the chance of a game over is high, but the payoff to Alice's curiosity will be high if she succeeds. If Alice is modelled as an episodic reinforcement learning agent, then her future return will be exactly zero if she gets a game over, which might make her overly risk averse. The real cost of a game over to Alice is the opportunity cost incurred by having to play through the game from the beginning (which is presumably less interesting to Alice having played the game for some time).

However using non-episodic returns for extrinsic rewards could be exploited by a strategy that finds a reward close to the beginning of the game, deliberately restarts the game by getting a game over, and repeats this in an endless cycle.

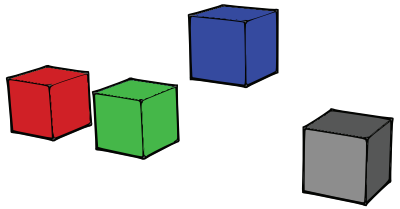
It is not obvious how to estimate the combined value of the non-episodic stream of intrinsic rewards i_t and the episodic stream of extrinsic rewards e_t. Our solution is to observe that the return is linear in

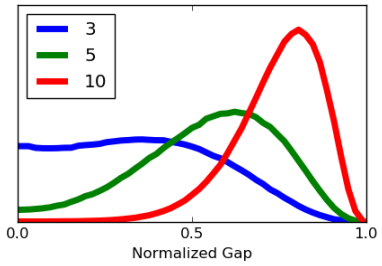
4

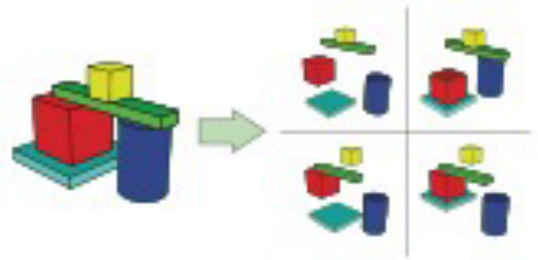
[1] *Yuri Burda et al.: Exploration by Random Network Distillation. ICLR 2019.*

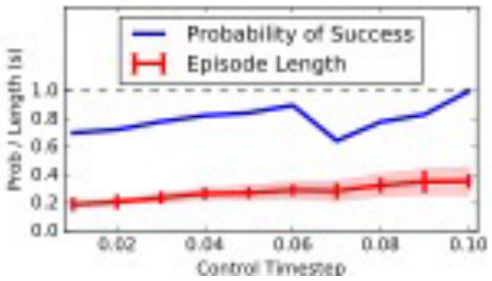
Physical Properties

- Motivation:
  - In many application (such as robotics) it helps the RL agent to explicitely understand and infer physical properties (such as mass, friction, etc.)
- Idea: Let an RL agent learn such properties by
  1. Exploration phase: letting the agent interact with the environment (without any specific task)
  2. Ask a question and give reward based on a labeling action (e.g.: *Which of the boxes is heaviest?*)
  - The agent must efficiently play around to figure out the physics and provide the correct answer
  - Exploration happens implicitly









[1] *Misha Denil et al.: Learning to Perform Physics Experiments via Deep Reinforcement Learning. ICLR 2017.*