Deep meta-RL and a way towards a general learner

Lajoie lab meeting September 1st

Léo Gagnon

Plan

- Introduction and motivation
  - A primer on reinforcement learning
  - Deep RL vs Humans
  - Sources of slowness of RL
- Meta RL
  - Problem formulation
  - Gradient-based methods
  - Black-box/Context-based methods
- Neuro-AI virtuous cycle
  - Prefrontal cortex as a meta-reinforcement learning system
  - Insights from Neuroscience and exemples
- Conclusion and research directions

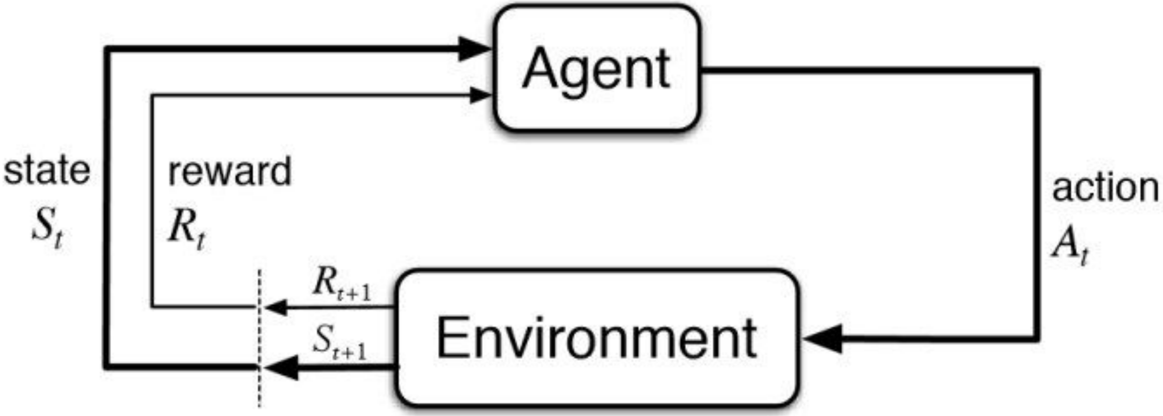
Introduction and motivation

Primer on Reinforcement Learning (RL)

Environment : A MDP defined by a the tuple $\{S, A, T, R\}$

Agent : Chooses an action every time-step according to a policy $\pi(a|s)$

Learning : Process that trains the policy to lead to maximum expected reward. Often, the estimation of a value function $V(s)$ with TD-learning plays an important role

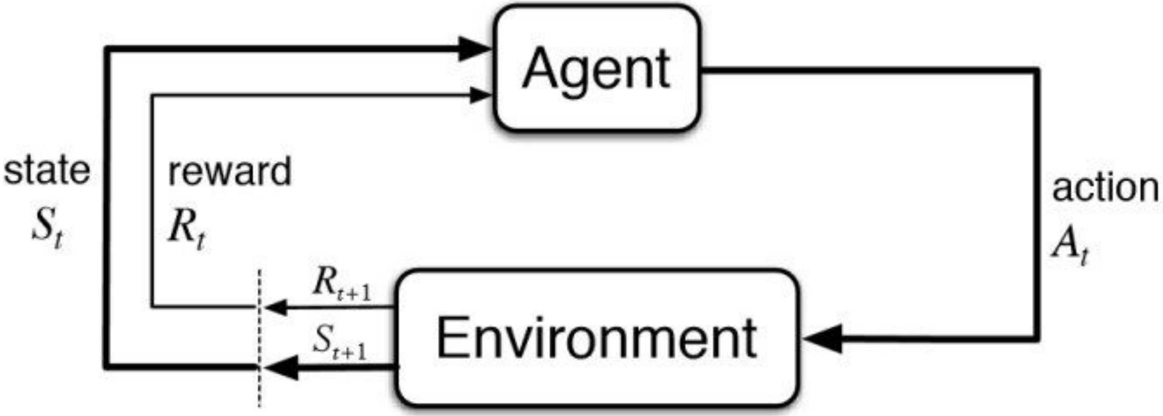


Primer on **Deep** Reinforcement Learning (DRL)

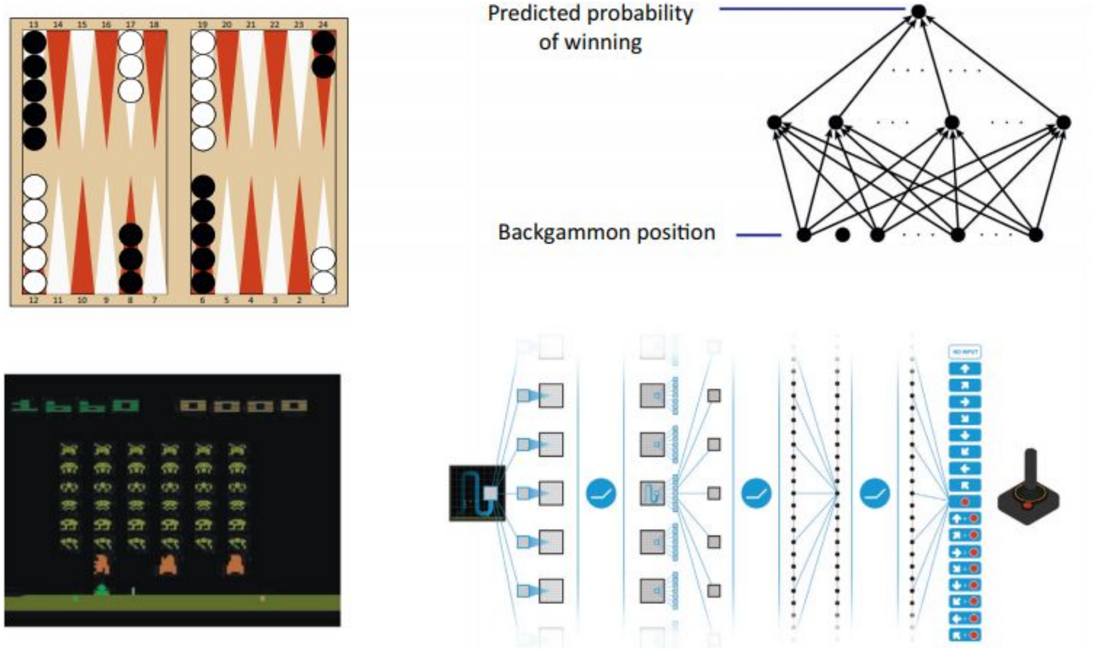
Environment : A MDP defined by a the tuple $\{S, A, T, R\}$

Agent : Chooses an action every time-step according to a policy $\pi_\theta(a|s)$

Learning : Process that trains the policy to lead to maximum expected reward. Often, the estimation of a value function $V_\theta(s)$ with TD-learning plays an important role



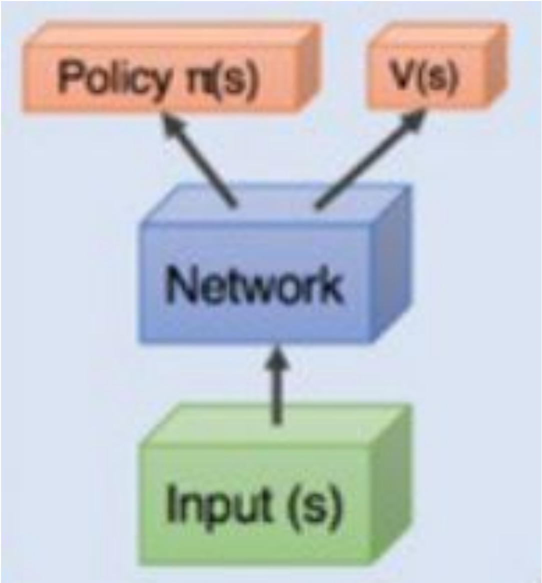
Primer on **Deep** Reinforcement Learning (DRL)



TD-Gammon and Q-Learning

Image from [Botvinick et al.]

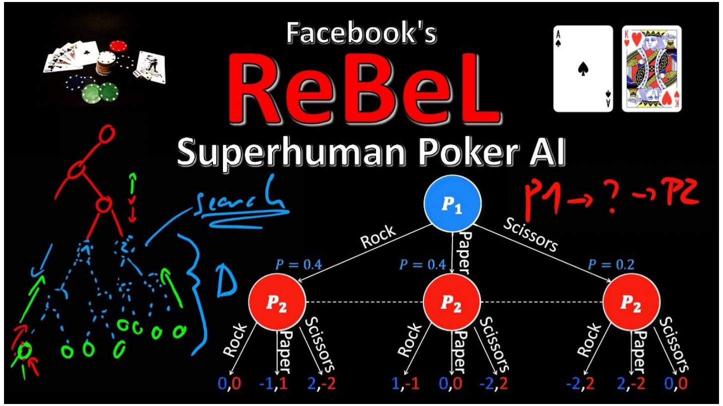
Primer on **Deep** Reinforcement Learning (DRL)

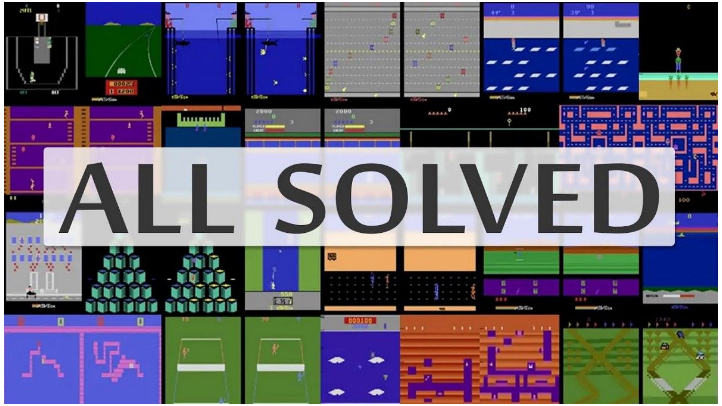


Actor Critic architectures
(A2C, A3C, IMPALA, …)

Image from [here]()

Deep RL successes

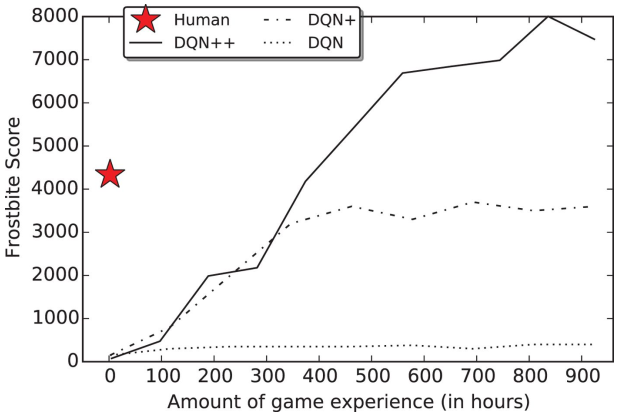




Deep RL vs Humans

While Deep RL has achieved impressive things, they come nowhere near the efficiency of human learning. In particular, they

- require massive volume of training data (sample inefficient)
- cannot adapt/generalize to new tasks

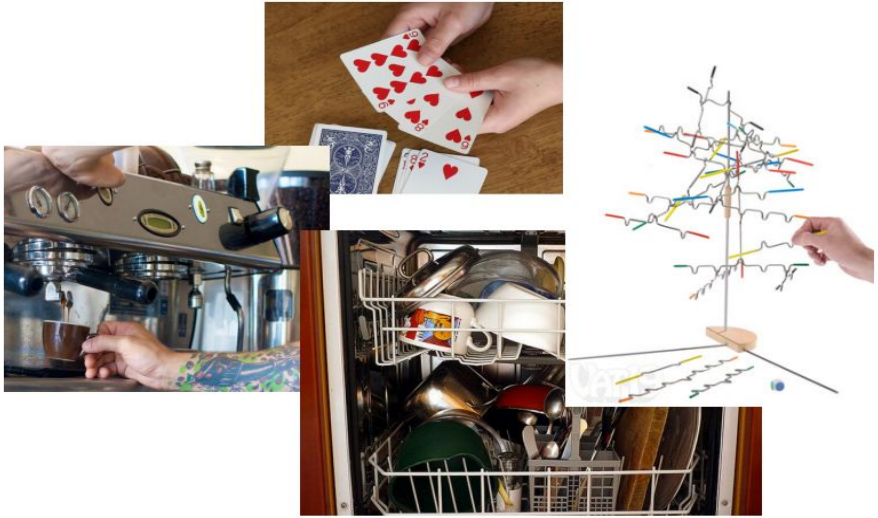


Deep RL vs Humans

Why are humans so efficient at learning?

Deep RL vs Humans

Why are humans so efficient at learning?



We learn structure common to wide range of tasks

Sources of the slowness of RL (Botvinick et al. 2019)

1. **Inductive biases**

   Any learning procedure necessarily faces a bias–variance trade-off: the stronger the initial assumptions the learning procedure makes about the patterns to be learned the less data will be required for learning to be accomplished. Classical DRL methods have almost no priors.

2. **Incremental parameter adjustment**

   DRL methods rely on gradient descent to learn. However, weights adjustments need to be small in order avoid overfitting and *catastrophic interference*. More generally, learning that directly maps perceptual inputs to actions HAS to be slow.

Sources of the slowness of RL (Botvinick et al.)

A general lesson to be learned is that

"[...] where fast learning occurs, it necessarily relies on slow learning, which establishes the representations and inductive biases that enable fast learning"

Digression : The Bitter Lesson

Q : If fast learning relies on already having useful biases and representation, and if the process of learning such priors is slow, then wouldn't be a good idea to build them in by hand?

A : "The actual contents of minds are tremendously, irredeemably complex; we should stop trying to find simple ways to think about the contents of minds, such as simple ways to think about space, objects, multiple agents, or symmetries. All these are part of the arbitrary, intrinsically-complex, outside world. They are not what should be built in, as their complexity is endless; instead we should build in only the meta-methods that can find and capture this arbitrary complexity. Essential to these methods is that they can find good approximations, but the search for them should be by our methods, not by us. We want AI agents that can discover like we can, not which contain what we have discovered. Building in our discoveries only makes it harder to see how the discovering process can be done." - Richard Sutton

Meta RL

Formulation of Meta RL

Regular RL: learn policy for single task

$$\theta^\star = \arg\max_\theta E_{\pi_\theta(\tau)}[R(\tau)]$$

$$= f_{\mathrm{RL}}(\mathcal{M})$$

MDP

Meta-RL: learn adaptation rule

$$\theta^\star = \arg\max_\theta \sum_{i=1}^{n} E_{\pi_{\phi_i}(\tau)}[R(\tau)]$$

$$\text{where } \phi_i = f_\theta(\mathcal{M}_i)$$

MDP for task i

$\mathcal{M}_1$ $\mathcal{M}_2$ $\mathcal{M}_3$

$\mathcal{M}_{test}$

Slide from here

Formulation of Meta RL

Regular RL: learn policy for single task

$$\theta^\star = \arg\max_\theta E_{\pi_\theta(\tau)}[R(\tau)]$$

$$= f_{\mathrm{RL}}(\mathcal{M})$$

MDP

Meta-RL: learn adaptation rule

$$\theta^\star = \arg\max_\theta \sum_{i=1}^{n} E_{\pi_{\phi_i}(\tau)}[R(\tau)]$$

where $\phi_i = f_\theta(\mathcal{M}_i)$

Meta-training / Outer loop

Adaptation / Inner loop

MDP for task i

$\mathcal{M}_1$ $\mathcal{M}_2$ $\mathcal{M}_3$

$\mathcal{M}_{test}$

Slide from [here]()

Formulation of Meta RL

$$\theta^\star = \arg\max_\theta \sum_{i=1}^{n} E_{\pi_{\phi_i}(\tau)}[R(\tau)]$$

$$\text{where } \phi_i = f_\theta(\mathcal{M}_i)$$

What should the adaptation procedure do?

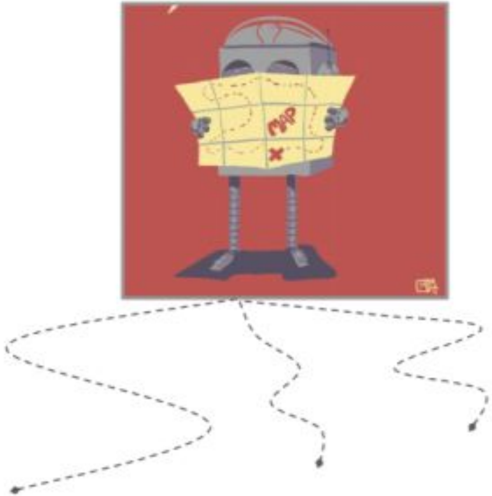
- **Explore**: Collect the most informative data
- **Adapt**: Use that data to obtain the optimal policy

Slide from here

Formulation of Meta RL

$$\theta^{\star} = \arg\max_{\theta} \sum_{i=1}^{n} E_{\pi_{\phi_i}(\tau)}[R(\tau)]$$

$$\text{where } \phi_i = f_{\theta}(\mathcal{M}_i)$$



What should the adaptation procedure do?

- **Explore**: Collect the most informative data
- **Adapt**: Use that data to obtain the optimal policy

Implement a “general” learner in the context of the tasks distribution.

Slide adapted from here

Formulation of Meta RL

Training procedure:

Repeat :
Outer loop (slow learning):

Sample new MDP, $M_i \sim \mathcal{M}$
Reset internal state of the fast learner

Inner loop (fast learning):

Compute adaptation $\phi_i = f_\theta(\mathcal{M}_i)$ with collected data $\mathcal{D}_i$

Update θ according to $\mathcal{L}(\mathcal{D}_i, \phi_i)$

Formulation of Meta RL

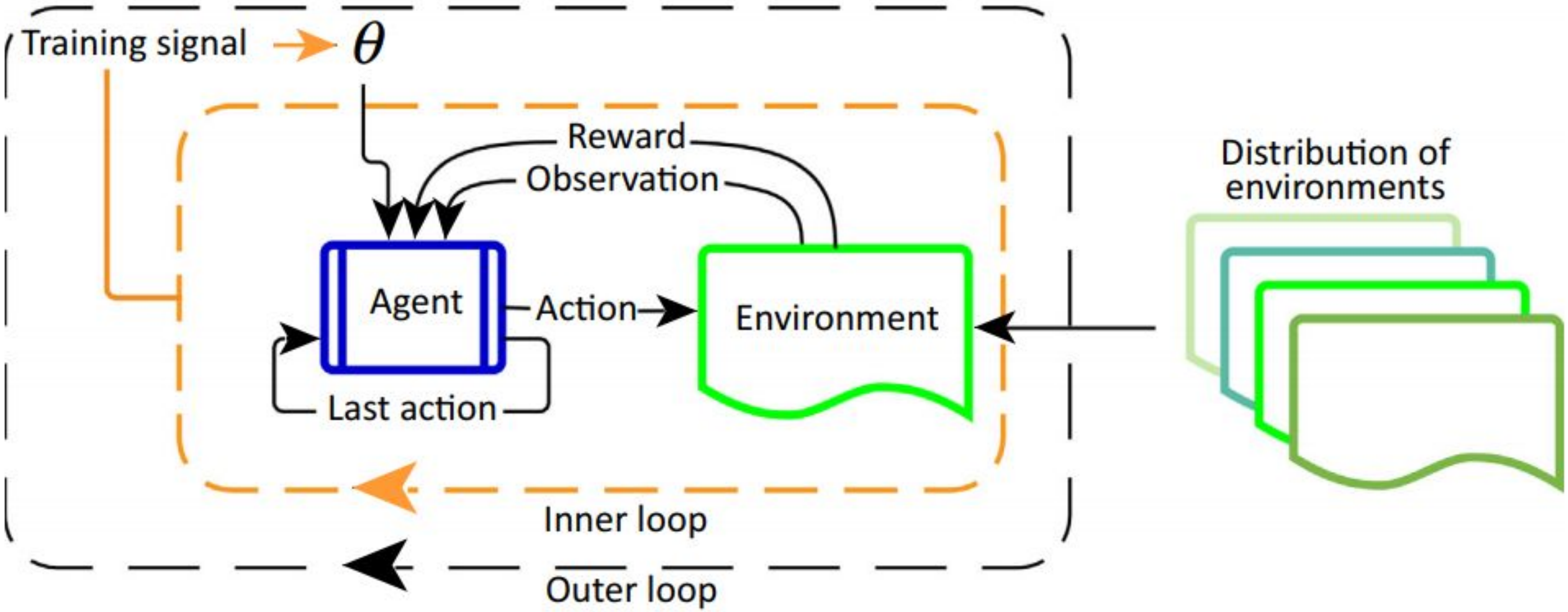


Image from here

Common approaches

Design choices for Meta RL come down to design choices for the fast learner. The two choices (that I know of) are :

1. A normal RL algorithm. What is meta-learned is the initialisation
2. A sequence modelling system implementing the whole learning algorithm in a black box manner

Gradient-based meta RL (MAML)

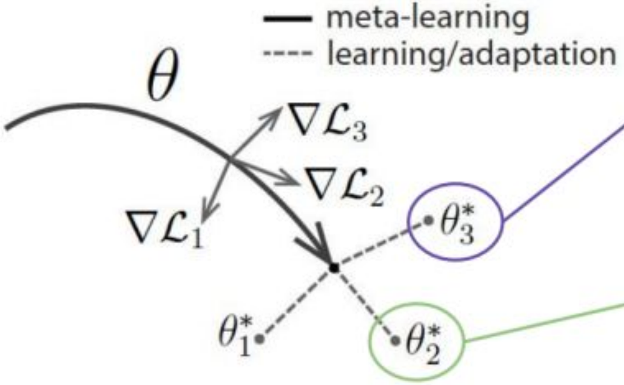
$$\theta^\star = \arg\max_\theta \sum_{i=1}^{n} E_{\pi_{\phi_i}(\tau)}[R(\tau)]$$

PG

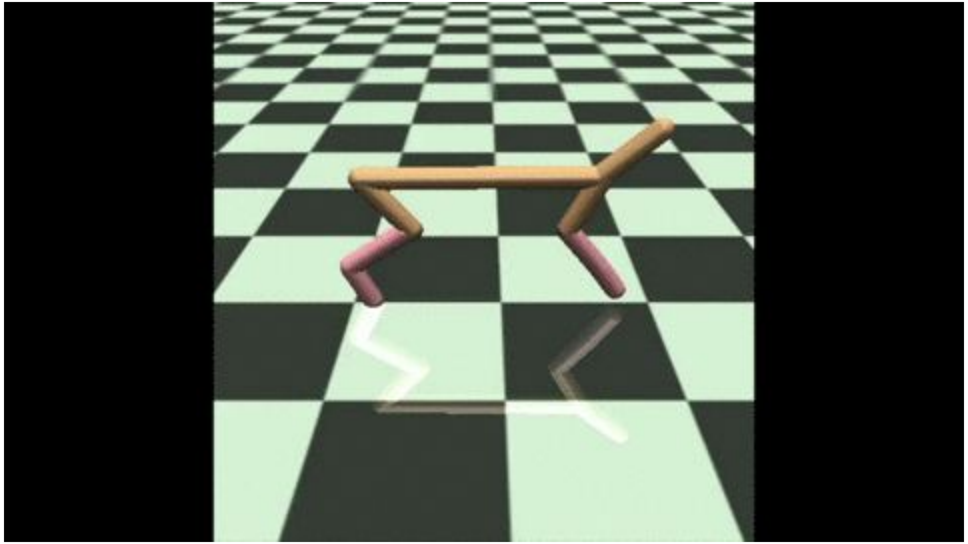
where $\phi_i = f_\theta(\mathcal{M}_i)$

PG

Learn a parameter initialization from which fine-tuning for a new task works!



Gradient-based meta RL (MAML)



Gradient-based meta RL (MAML)

while training:
  for i in tasks:
  1. sample k episodes $\mathcal{D}_i = \{(s, a, s', r)\}_{1:k}$ from π_θ
  2. compute adapted parameters $\phi_i = \theta - \alpha \nabla_\theta \mathcal{L}_i(\pi_\theta, \mathcal{D}_i)$
  3. sample k episodes $\mathcal{D}'_i = \{(s, a, s', r)_{1:k}\}$ from π_ϕ

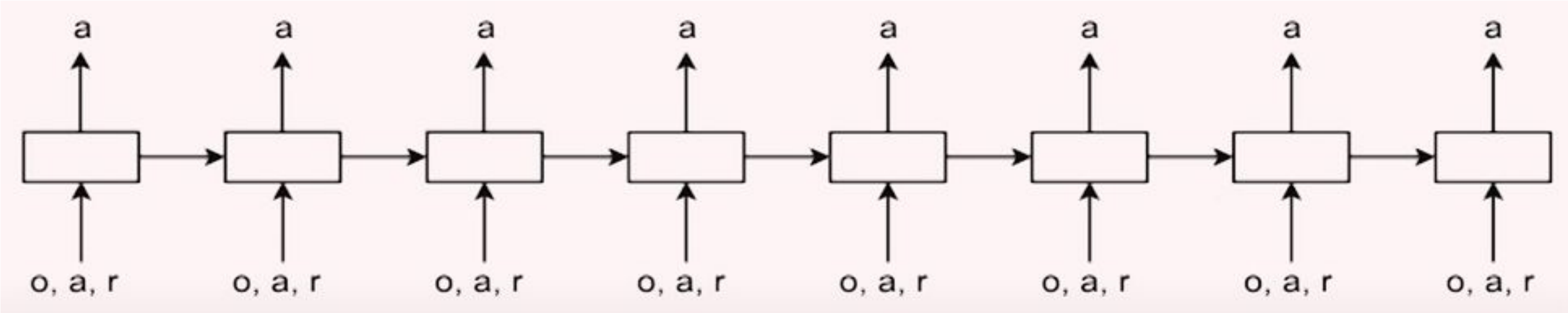
update policy parameters $\theta \leftarrow \theta - \nabla_\theta \sum_i \mathcal{L}_i(\mathcal{D}'_i, \pi_{\phi_i})$

Requires second order derivatives!

Sidestep the "small incremental steps" by pretraining the learner so that there is not much to learn.

Black-box methods

Reinforcement learning, but implemented in the dynamics of the RNN!



Black-box methods

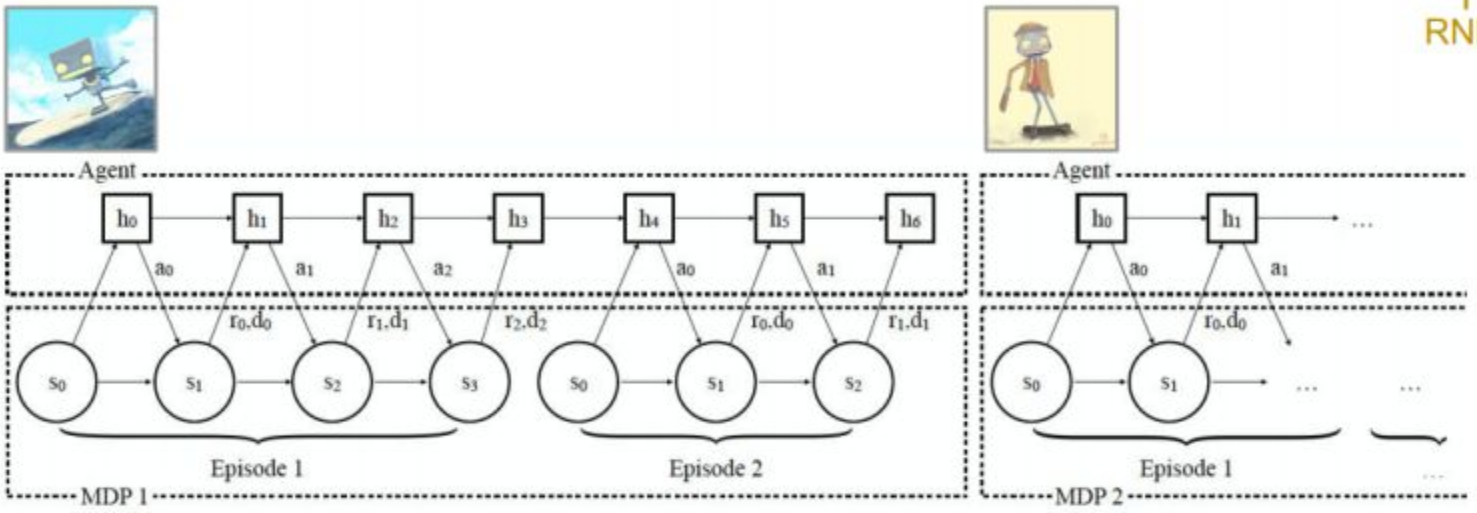
$$\theta^\star = \arg\max_\theta \sum_{i=1}^{n} E_{\pi_{\phi_i}(\tau)}[R(\tau)]$$

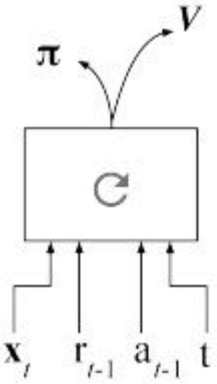
PG

$$\text{where } \phi_i = f_\theta(\mathcal{M}_i)$$

RNN

Implement the policy as a recurrent network, train across a set of tasks





Black-box methods

How is that different from a standard recurrent policy?

1. The recurrent model takes also as input the past reward, past action and time-step.
2. The hidden state is conserved between episodes

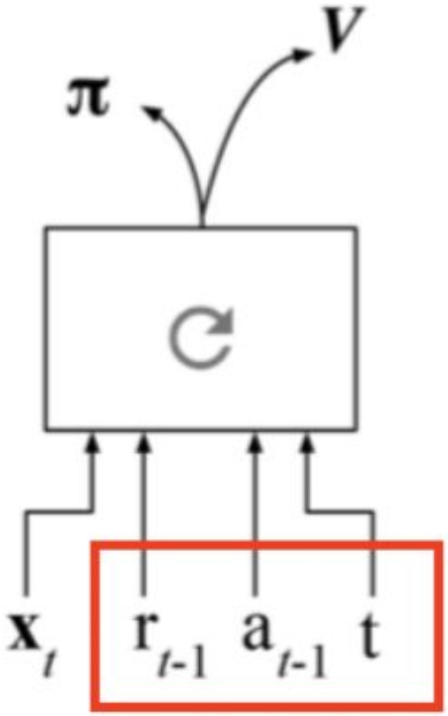


Image from Wang et al. 2017

Black-box methods

while training:
  for i in tasks:
    initialize hidden state $\mathbf{h_0} = 0$
    for t in timesteps:

1. sample 1 transition $\mathcal{D}_i = \mathcal{D}_i \cup \{(s_t, a_t, s_{t+1}, r_t)\}$ from π_{h_t}
2. update policy hidden state $\mathbf{h_{t+1}} = f_\theta(\mathbf{h_t}, s_t, a_t, s_{t+1}, r_t)$

update policy parameters $\theta \leftarrow \theta - \nabla_\theta \sum_i \mathcal{L}_i(\mathcal{D}_i, \pi_\mathbf{h})$

Sidestep the “small incremental steps” by only operating neural activity/representations.

Black-box methods

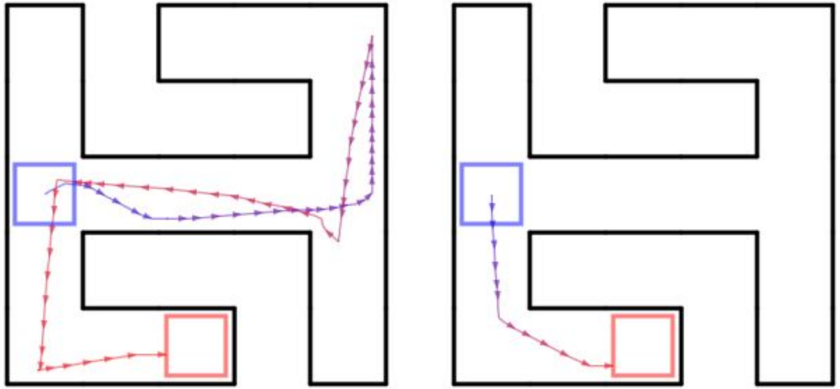


Image from [Duan et al. 2017]

Black-box methods

The learning procedure of the fast learner is different from the meta-learning algorithm and it is tailored by the structure of the tasks. For example, Wang et al. (2016) showed that it performs better on correlated bandits than on independent ones.

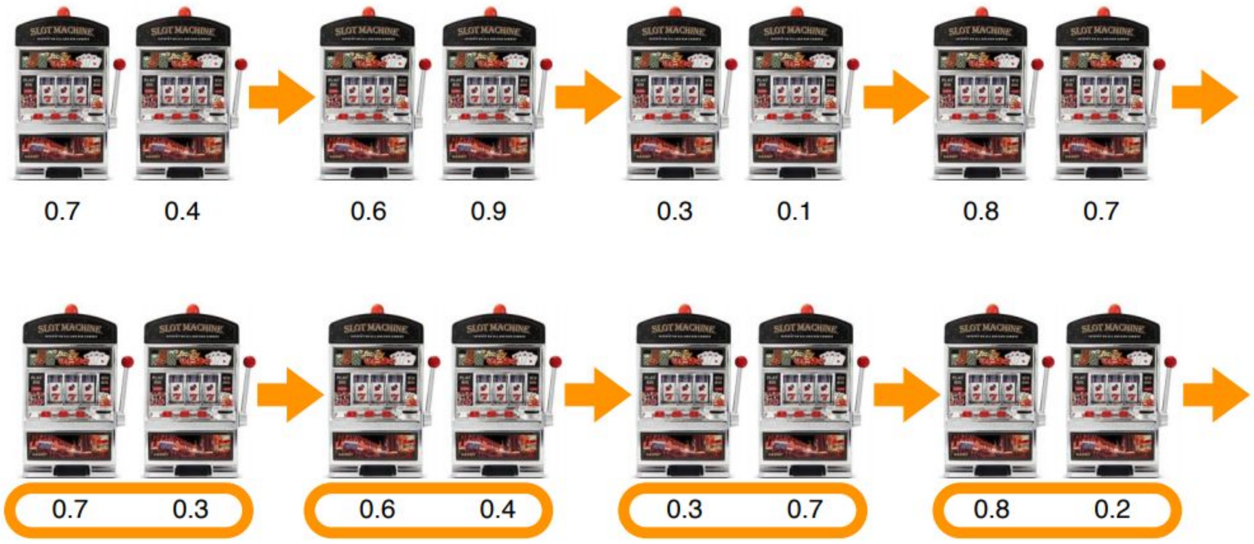


Image from Wang et al. 2018

Black-box methods

Hidden state dynamics of the inner learner for the correlated bandits tasks shows that the learner has meta-learned the structure of the environment.

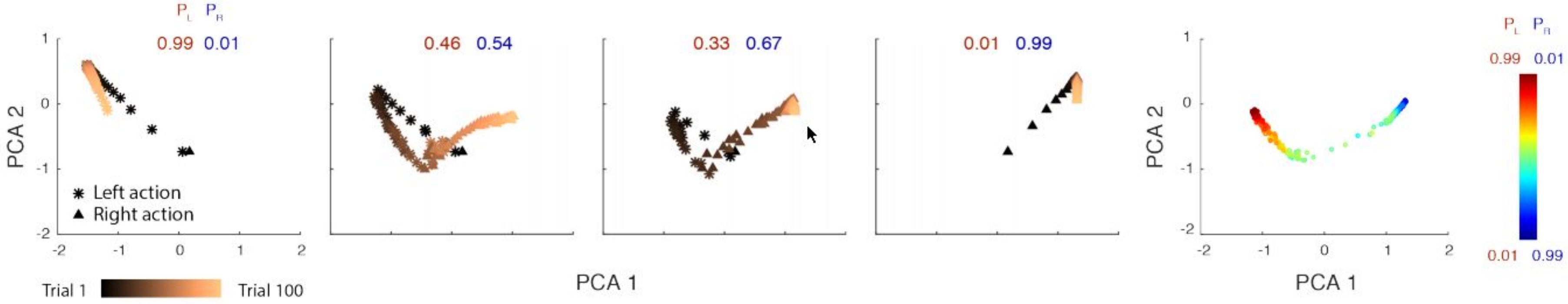


Image from [Wang et al. 2018]

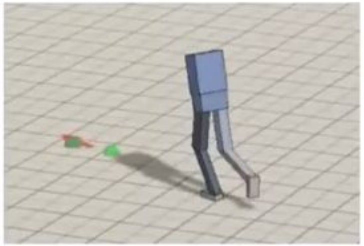
Reframing with contextual policy and belief state

To get a better sense of what this method is doing, it is helpful to think of the inner learner as slow-learning a contextual policy where the context is a function of the experience.

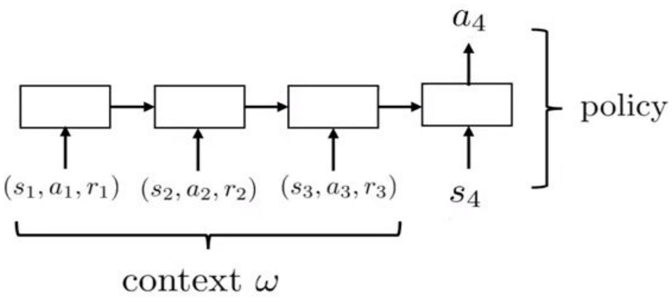
$$\pi_\theta(a|s, \omega)$$

The fast-learning would consists in retrieving ω from the environment and then acting on it. This context can be conceived as what the learner believes the task/goal is (within the distribution), perhaps in a distributed way.

ω: stack location



ω: walking direction



Leveraging the contextual policy insight

Learning a full learning procedure end-to-end is hard!

In particular, because the training signal only tells the inner learner to maximise reward, the exploration-exploitation is hard.

The main way people have tackled this problem is by using the contextual policy framework and by building in the fast learner a task-inference phase.

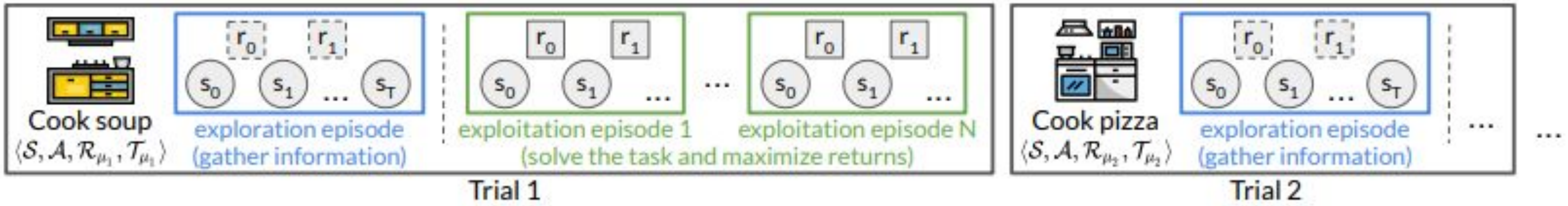
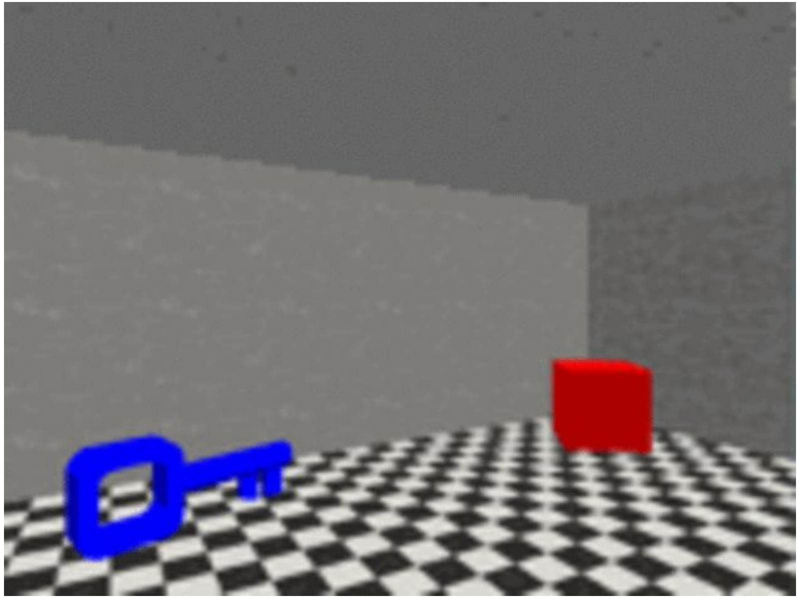
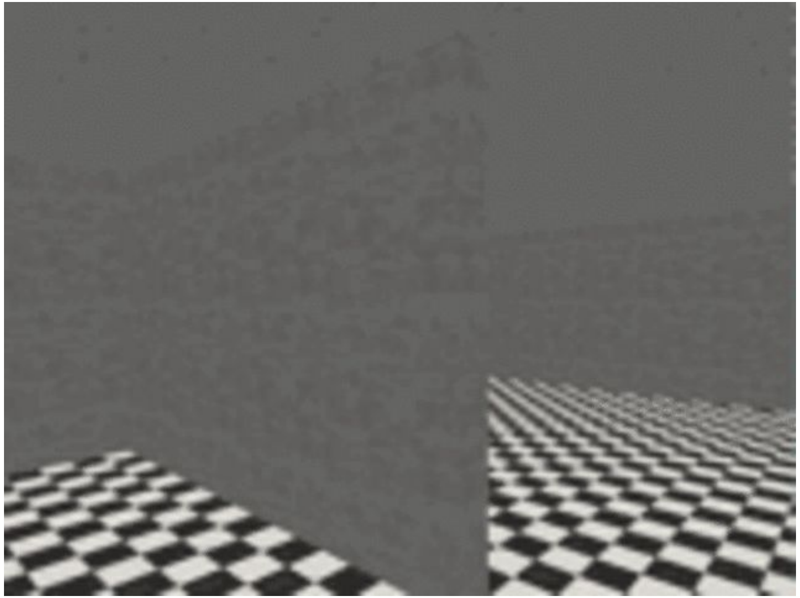


Image from DREAM

Leveraging the contextual policy insight





Gifs from [DREAM]

Neuro-AI virtuous cycle

See this

Neuro-AI virtuous cycle (Hassabis et al. 2017)

Neuroscience and Artificial Intelligence can create a “virtuous cycle” advancing the objectives of both fields : understanding intelligence.

Neuroscience offers inspiration and confirmation and AI is easier to study.

Exemple :

Ideas from animal psychology → RL → TD-learning in the brain

(O’Doherty et al., 2003, Schultz et al., 1997)

Prefrontal cortex as a meta-reinforcement learning system (Wang et al. 2018)

The standard model of reward-learning in the brain goes as follows :

“Phasic dopamine (DA) release is interpreted as conveying a reward prediction error (RPE) signal, an index of surprise which figures centrally in temporal-difference RL algorithms. Under the theory, the RPE drives synaptic plasticity in the striatum, translating experienced action-reward associations into optimized behavioral policies.”

However, recent studies found that neural activity in the PFC appears to reflect a set of operations that together constitute a self-contained RL algorithm.

Wang et al. 2018 propose a new theory of reward-based learning in the brain following meta-RL.

Prefrontal cortex as a meta-reinforcement learning system (Wang et al. 2018)

The new formulation goes as follows :

- System architecture

  The PFC, along with other connected structures, form a recurrent neural network. This network inputs perceptual data and outputs action commands and estimates of state value.

- Learning

  Synaptic weights in the PFN are adjusted by a model-free RL procedure in with DA conveys the RPE signal. DA contributes to slow learning, PFN dynamics to fast learning.

- Task environment

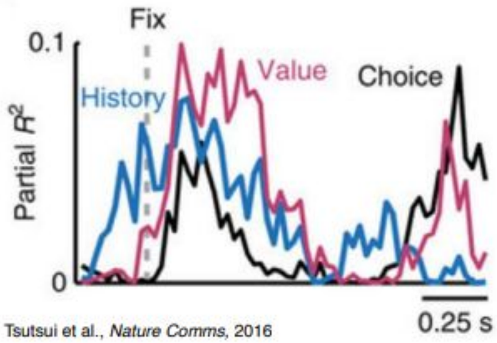
  Learning takes place in a dynamic environment posing a series of interrelated tasks.

Prefrontal cortex as a meta-reinforcement learning system (Wang et al. 2018)

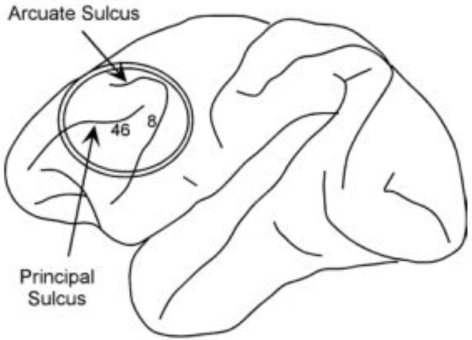
The paper consists of 6 simulations where they match experimental neuroscience findings to Meta-RL models outputs.

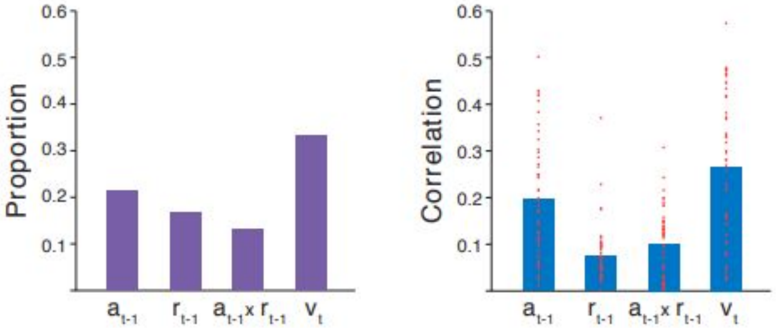
1. Reinforcement learning in the prefrontal network
2. Adaptation of prefrontal-based learning to the task environment
3. Reward prediction errors reflecting inferred value
4. ‘Model-based’ behavior: The Two-Step Task
5. Learning to learn
6. The role of dopamine: Effects of optogenetic manipulation

Simulation 1 : Reinforcement learning in the prefrontal network

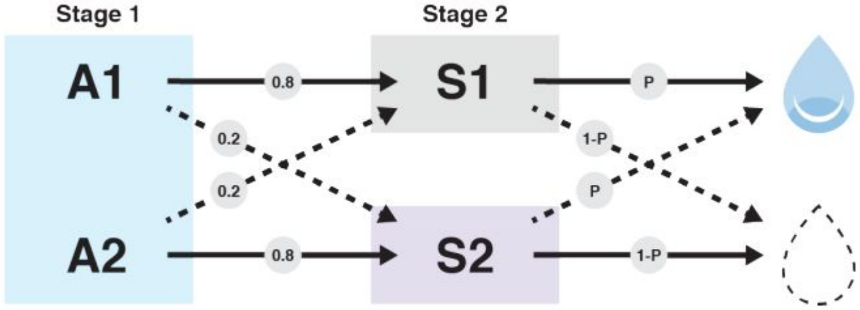


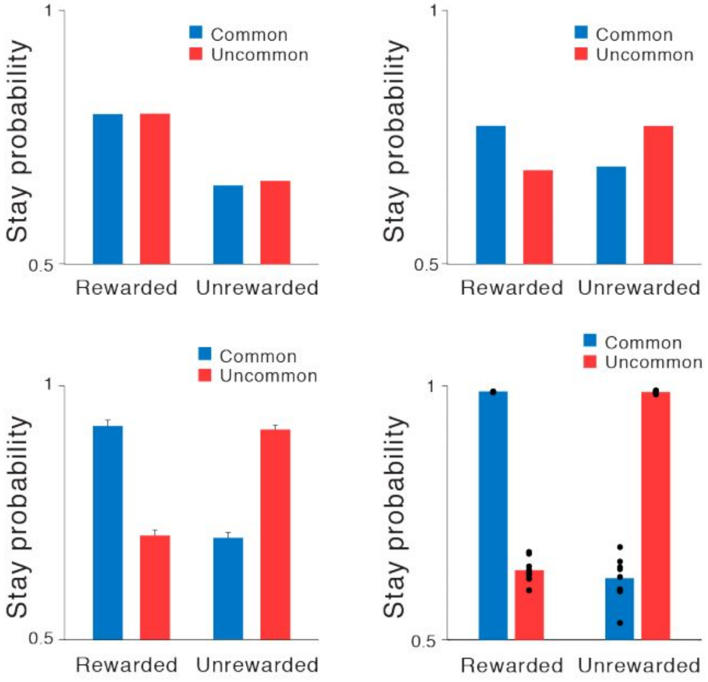
Tsutsui et al., *Nature Comms*, 2016



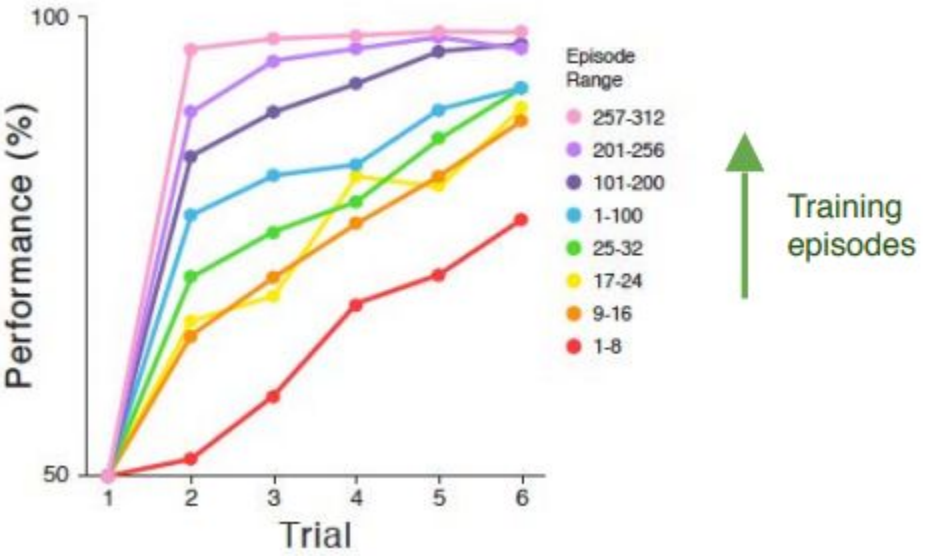


Simulation 4 : ‘Model-based’ behavior: The Two-Step Task

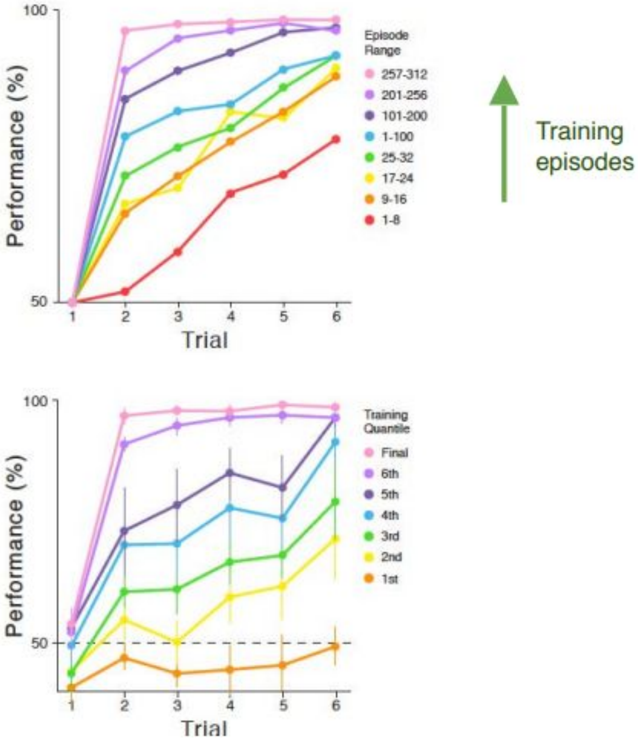




Simulation 5 : Learning to learn

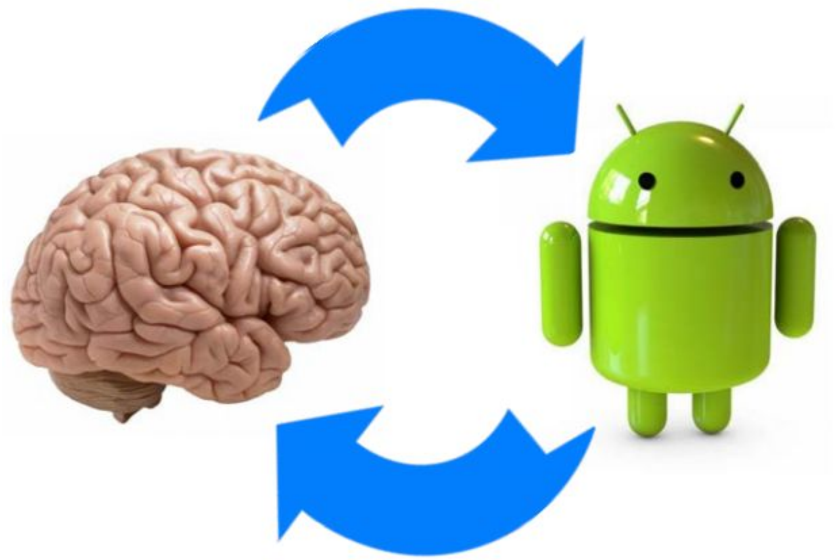


Simulation 5 : Learning to learn



Neuro-AI virtuous cycle

Knowing that the brain probably does Deep Meta-RL, we can use knowledge from neuroscience and psychology to inform design decisions for our Deep Meta-RL systems.

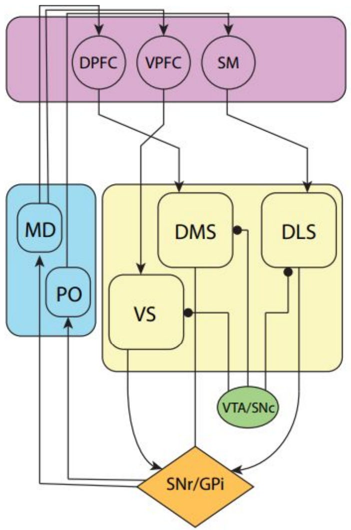


Neuro-AI virtuous cycle

Insight 1 : Architecture

The recurrent model implementing the fast learner in the brain (the PFN) is WAY more complex and structured than a simple LSTM. Here are some differences :

- The PFN contains distinct regions (modularity), cortical layers and cell types.
- Some regions process information hierarchically, some in parallel.
- Some regions are architecturally or functionally very different from the cortex (basal ganglia, thalamus, hippocampus)

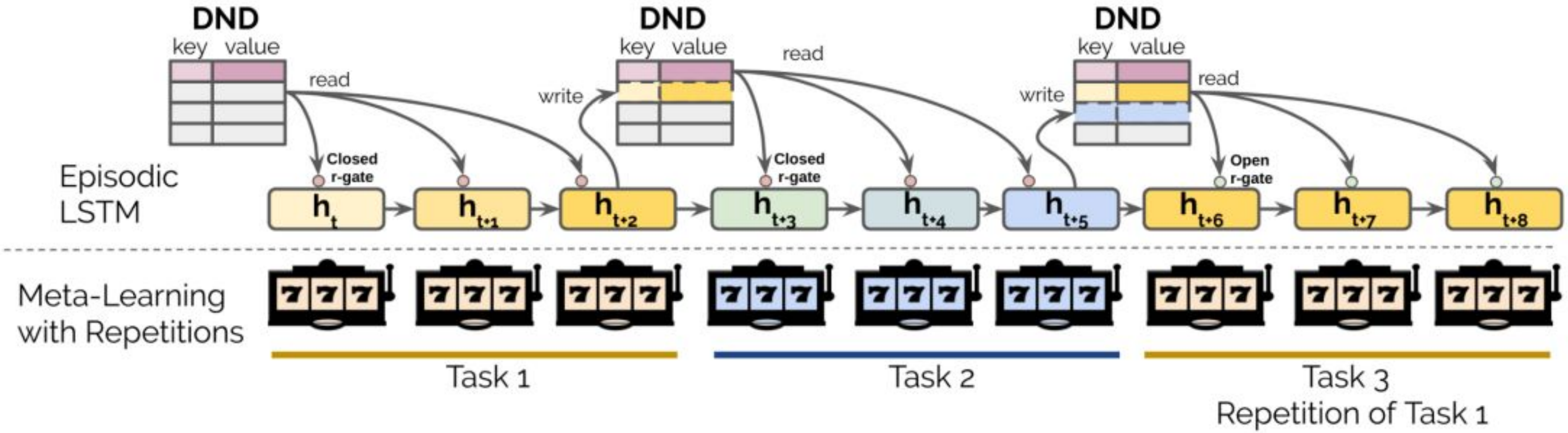


Ideas inspired from [Yang et al. 2021]

Neuro-AI virtuous cycle

Insight 1 : Architecture (Exemple)

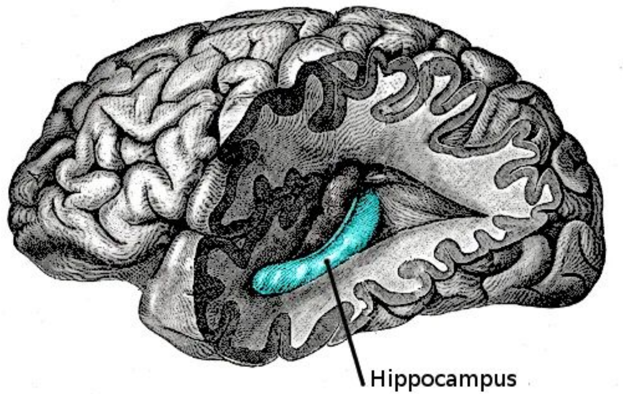
“Inspired in part by evidence that human episodic memory retrieves past working memory states”, [Ritter et al. (2018)] improved the L2RL architecture with a episodic memory module.



Neuro-AI virtuous cycle

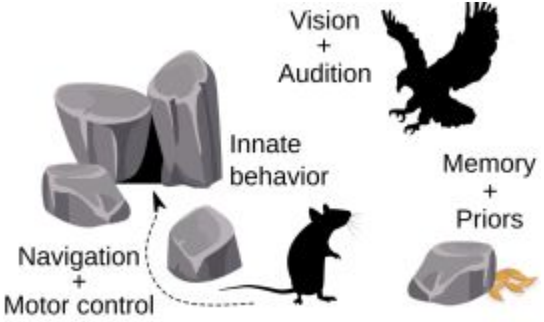
Insight 1 : Architecture (Exemple)

“The key takeaway from the success so far of EMRL is a proof of the sufficiency of a small set of well motivated **architectural components**, when trained to optimize a specific objective function, to produce a variety of episodic and incremental learning processes observed in humans.”
-Ritter et al. 2018



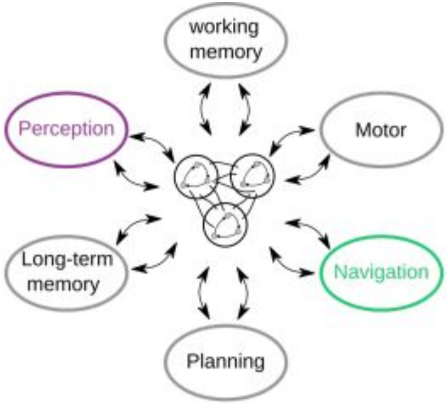
Neuro-AI virtuous cycle

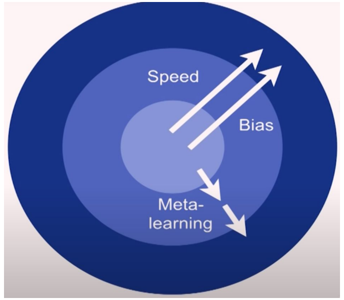
Insight 2 : Evolution, development and naturalistic tasks

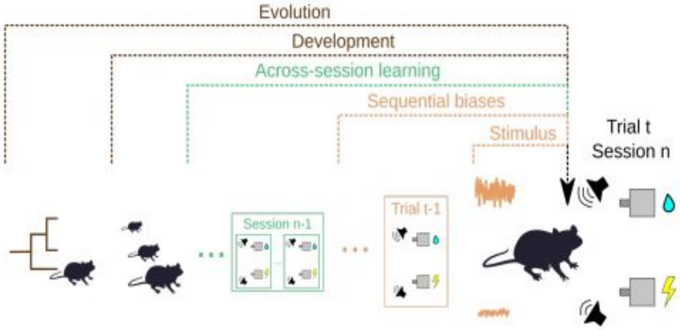


Animals learning includes much broader task distribution and learning timescale than current Meta-RL.

- Animals arrive at the laboratory with priors built by evolution and developmental period (unsupervised learning, curriculum learning)
- Animals and humans train on a VERY broad distribution of naturalistic tasks. This forces the agent to learn underlying structure of the world







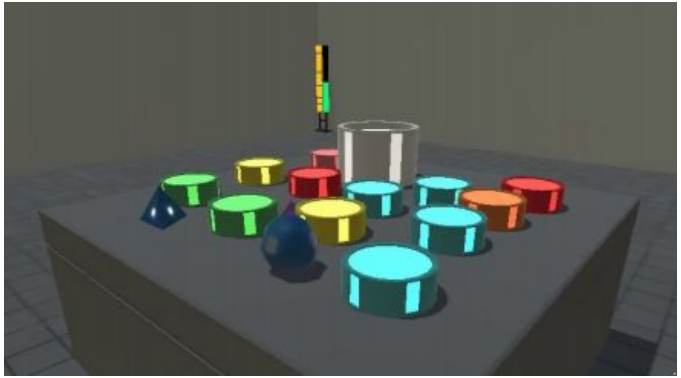
Ideas inspired from [Yang et al. 2021]

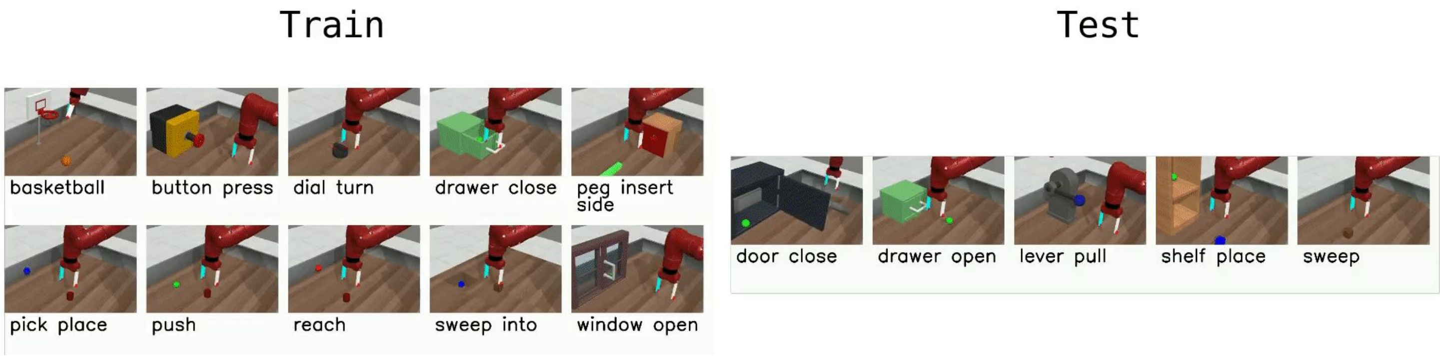
Neuro-AI virtuous cycle

Insight 2 : Evolution, development and naturalistic tasks (Exemple)

Recently two large scale Meta-RL task distributions have been introduced

- Meta-world : 50 distinct robot manipulation environments
- Alchemy : A complex combinatorial game in Unity 3D





Digression : The Bitter Lesson (bis)

Supposing that the brain in doing Meta-RL (and very well), having access to its architecture/morphology is very precious. Perhaps if we build a system with similar components (functionally), a similar kind of intelligence can emerge.

It enables us to go from architecture to behavior instead of the inverse, which seems like a better idea for scalability.

Conclusion and research direction

Deep Meta-RL and a way towards a general learner

The things to remember about this presentation

Deep Meta-RL and a way towards a general learner

The things to remember about this presentation

- Classical RL is doomed to be slow and generalize poorly because it lacks inductive priors and its learning speed is limited by gradient descent
    - The Bitter Lesson : Building in the priors is not a good idea

Deep Meta-RL and a way towards a general learner

The things to remember about this presentation

- Classical RL is doomed to be slow and generalize poorly because it lacks inductive priors and its learning speed is limited by gradient descent
    - The Bitter Lesson : Building in the priors is not a good idea
- Deep Meta-RL offers a way to do fast RL by slowly learning inductive biases

Deep Meta-RL and a way towards a general learner

The things to remember about this presentation

- Classical RL is doomed to be slow and generalize poorly because it lacks inductive priors and its learning speed is limited by gradient descent
  - The Bitter Lesson : Building in the priors is not a good idea
- Deep Meta-RL offers a way to do fast RL by slowly learning inductive biases
- The fact that brain (probably) implements Meta-RL makes insights from Neuroscience and Psychology precious to Meta-RL research in AI

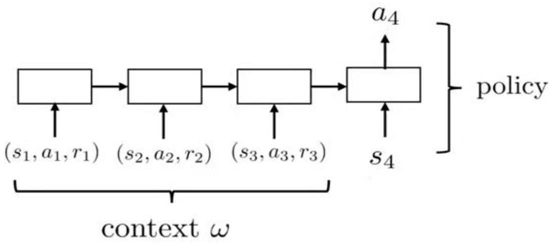
Deep Meta-RL and a way towards a general learner

The things to remember about this presentation

- Classical RL is doomed to be slow and generalize poorly because it lacks inductive priors and its learning speed is limited by gradient descent
    - The Bitter Lesson : Building in the priors is not a good idea
- Deep Meta-RL offers a way to do fast RL by slowly learning inductive biases
- The fact that brain (probably) implements Meta-RL makes insights from Neuroscience and Psychology precious to Meta-RL research in AI
- By concentrating our efforts on architecture and capacity, it is conceivable that a Meta-RL agent becomes as general a learner as the task distribution needs it to be (scalability) with enough time

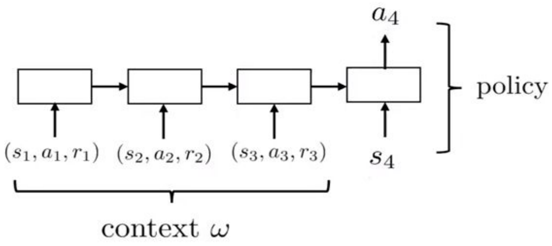
Research directions

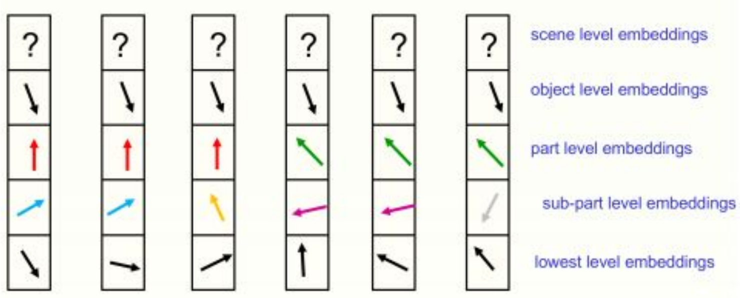
- Thinking in terms of contextual policy, enforce a part of the RNN to meta-learn general task primitives (tailored to the task distribution) and other one to combine (with attention) such primitive computation based on context.



Research directions

- Thinking in terms of contextual policy, enforce a part of the RNN to meta-learn general task primitives (tailored to the task distribution) and other one to combine (with attention) such primitive computation based on context.
- Inspired by the architecture in GLOM, constantly update an uncertain predictive representation of ω based on experience. Acting based on this uncertain representation might lead to satisfying explore-exploit mechanism.





Research directions

- Thinking in terms of contextual policy, enforce a part of the RNN to meta-learn general task primitives (tailored to the task distribution) and other one to combine (with attention) such primitive computation based on context.
- Inspired by the architecture in GLOM, constantly update an uncertain predictive representation of ω based on experience. Acting based on this uncertain representation might lead to satisfying explore-exploit mechanism.
- And more : Distributional RL, Hierarchical RL, RMCs, RIMs, GVFs