# 1 Introduction

1. Current RL algorithms are lacking in two ways:
2. Too sample inefficient where as human learners can attain reasonable performance on any of a wide range of tasks without comparatively little experience.
3. Usually specialize in one domain, whereas human learners can flexibly adapt to changing task conditions.
4. The key concept is to use standard RL techniques to train a RNN such that the RNN implements its own, free-standing RL procedure.

# 2 Methods

## 2.1 Background

1. Flexible, data-efficient learning requires prior biases, which can be engineered directly into the learning systems or acquired through learning.
2. In a standard setup, the learning agent is confronted with a series of tasks that different from one another, but share some underlying set of regularities.
3. Meta-learning is then defined as an effect where the learning agent improves its performance on each new tasks more rapidly, on average, than in past tasks.
4. A very relevant work is by Hochreiter 2001, where a RNN is trained on a series of related tasks using backprop.
5. A critical aspect of their setup is the RNN receives, at each step within a task, an auxiliary input indicating the target output for the preceding step.
6. This sentence is cryptic: "In this scenario, a different function is used to generate the data in each training episode, but if the functions are all drawn from a single parametric family, then the system gradually tune into this consistent structure, converging on accurate outputs more and more rapidly across episodes."
7. The interesting aspect is that the process underlying learning within each task lies entirely within the dynamics of the RNN.
8. They showed that after an initial training period, the network can improve its performance on new tasks even if the weights are held constant.

## 2.2 Deep Meta-RL background

1. Hochreiter's work was in supervised learning.
2. This paper considers it in the context of RL.
3. Rather than presenting target outputs as auxiliary inputs, the agent receives inputs indicating the action output and the reward resulting from that action on the previous step.
4. They emphasize that the dynamics of the learnt RNN implements a learning algorithm entirely separate from the one used to train the RNN's weights.
5. They emphasize that the learned RL procedure inside the RNN's weights can differ starkly from the algorithm used to train the RNN's weights.

### 2.3 Formalism

1. Let $\mathcal{D}$ be a distribution over MDP.
2. They want to demonstrate that meta-RL can learn a prior-dependent RL algorithm, in the sense that it will perform well on average on MDPs drawn from $\mathcal{D}$.
3. The agent is represented by a RNN.
4. At the start of a new episode, a new MDP task $m \sim \mathcal{D}$ and an initial state for this task are sampled.
5. The internal state of the RNN is reset.
6. The agent then interacts with the MDP for a number of discrete time-steps.
7. At each step $t$, the action $a_t$ is a function of the whole history $\mathcal{H}_t = \{x_0, a_0, r_0, \ldots, x_{t-1}, a_{t-1}, r_{t-1}, x_t\}$
8. The RNN's weights are trained to maximize the sum of observed rewards over all steps and episodes.
9. After training, the RNN's weight is fixed, and it is evaluated on a set of MDPs sampled from the same distribution $\mathcal{D}$ or similar distribution.

## 3 Experiments

1. They consider 6 proof-of-concepts experiments.
2. The first thing they are interested in is to see if meta-RL can learn an adaptive balance between exploration and exploitation.
3. The second thing they are interested in is whether meta-RL can give rise to learning that gains efficiency by capitalizing on task structure.

### 3.1 Bandit problems

1. They demonstrate that meta-RL can learn prior-dependent bandit algorithm.
2. The meta-RL system is trained on a sequence of bandit environments through episodes.
3. At the start of a new episode, its RNN state is reset and a bandit task is sampled.
4. A bandit task is defined as a set of distributions, one for each arm, from which rewards are sampled.
5. The agent plays in this bandit environment for a number of trials and is trained to maximize observed rewards.
6. After training, the agent's policy is evaluated on a set of bandit tasks that are drawn from a test distribution.
7. They evaluate the performance of the learned bandit algorithm by the cumulative regret, a measure of the loss in expected rewards suffered when playing sub-optimal arms.
8. Let $\mu_a(b)$ be the expected reward of arm $a$ in bandit environment $b$.
9. Let $a^*(b)$ be one optimal arm.
10. Let $\mu^*(b) = \max_a \mu_a(b) = \mu_{a^*(b)}(b)$ the optimal expected reward.
11. The cumulative regret in environment $b$ is defined to be $R_T(b) = \sum_{t=1}^{T} \mu^*(b) - \mu_{a_t}(b)$, where $a_t$ is the arm chosen at time $t$.
12. The performance is averaged over bandit environments drawn from the test distribution, either in terms of the cumulative regret $\mathbb{E}_{b\sim\mathcal{D}'}\left[R_T(b)\right]$ or in terms of the number of suboptimal pulls $\mathbb{E}_{b\sim\mathcal{D}'}\left[\sum_{t=1}^{T} \mathbb{I}\left\{a_t \neq a^*(b)\right\}\right]$.

### 3.1.1 Bandit with independent arms

1. They consider a simple two-armed bandit task to examine the behavior of meta-RL under conditions where theoretical guarantees exist and general purpose algorithms apply.

2. They demonstrate that meta-RL can roughly match the performance of algorithms specifically designed for this problem.

3. To verify the importance of passing the reward information the RNN, they removed this input and the performance was at chance levels on all tasks.

### 3.1.2 Bandit with dependent arms

1. A key hypothesis about meta-RL is that it gives rise to a learned RL algorithm that exploits consistent structure in the training distribution.

2. This experiment aims to gather empirical evidence for this hypothesis.

3. They consider Bernoulli distribution where the parameters $(p_1, p_2)$ of the two arms are correlated in the sense that $p_1 = 1 - p_2$. They consider several type of distributions:

4. Uniform: $p_1 \sim \mathcal{U}([0, 1])$

5. Easy: $p_1 \sim \mathcal{U}(\{0.1, 0.9\})$ (uniform distribution over the two possible values)

6. Medium: $p_1 \sim \mathcal{U}(\{0.25, 0.75\})$

7. Hard: $p_1 \sim \mathcal{U}(\{0.4, 0.6\})$

8. They verify that easy task is easier to learn than the medium which is easier than the hard task.

9. This is compatible with the notation that the hardness of a bandit problem is inversely proportional to the difference between the expected reward of the optimal and sub-optimal arm.

10. They note that withholding the reward information from the LSTM result in chance performance in even the easy task.

11. They demonstrate experimentally that meta-RL gives rise to learnt RL algorithm that can take advantage of structure in the training distribution.

12. The learnt RL algorithm can also generalize to a test distribution with different structure.

### 3.1.3 Bandit with restless arms

1. In previous bandit tasks, the problems were stationary and the agent's actions yielded information about task parameter that remained fixed throughout the episode.

2. This bandit has reward probabilities that changes within each episode.

3. In a low volatility setting, the probability changes slow. In high volatility setting, the probability changes faster.

4. The perform well, the agent needs to infer the volatility of the episode and adjust its own learning rate accordingly.

5. In high volatility environment, learning rate should be higher when the environment is changing rapidly, because past information becomes irrelevant more quickly.

6. They show that meta-RL outperforms traditional methods.

7. Bayesian Information Criterion shows that meta-RL's behavior was strongly related to the volatility of the episodes, indicating that meta-RL adjusts its learning rate to the volatility.

## 3.2 Markov Decision Problems

### 3.2.1 Two step tasks

1. A task used in neuroscience lit to distinguish the contribution of different systems viewed to support decision making.

2. Developed to dissociate model-free to model-based RL

3. The demonstrated that the meta-RL give rise to a model-based strategy, despite the use of model-free algorithm to train the RNN's weights.

### 3.2.2 Learning abstract task structure

1. They want to study the scalability of meta-RL using a task with rich visual inputs, long time horizons and sparse rewards.

2. In this task, a series of objects play defined roles which the system must infer.

3. They adapted Harlow's experiments to the visual domain and demonstrate that meta-RL reflects an impressive form of one-shot learning.