TED UNIVERSITY

CORE. Lab.

Review

From Reinforcement Learning to Cognitive Psychology: with Navigational Strategies in Mind

Firstname Lastname [1], Firstname Lastname [1], Firstname Lastname [2] and Firstname Lastname [2,*]

[1] TED University, Department of Mechanical Engineering, Ankara, Turkey; e-mail@tedu.edu.tr
[2] Affiliation 2; e-mail@e-mail.com
* Correspondence: e-mail@tedu.edu.tr; Tel.: (optional; include country code; if there are multiple corresponding authors, add author initials)

Abstract: Although machine Reinforcement Learning (RL) was developed with direct intuition from animal learning, it was not initially intended to explain the neurobiological processes of the brain. But the collaboration of neuroscientists, roboticists and computer scientists has recently permitted to draw parallels between the computational steps of RL algorithms and the cognitive processes in the brain. This system equivalence became even more evident after the invention of recurrent neural networks (RNNs) like LSTM and Transformer networks, which seem to fulfill the function of short-term memory in the brain's hippocampus. In this paper, we outline the relevance of RL in cognitive behavior especially in the context of action selection, spatial memory and navigational strategies. We are supporting our conclusions by mathematical models of the involved brain regions and the interplay of the neuromodulators that are engaged in these cognitive tasks. We also present the recent findings about Meta-Learning as the leading RL class of algorithms which came as the most tenable solution to the problem of sample inefficiency in generic RL. Therefore, Meta-Learning and Meta-RL models offer the closest match and the simplest explanation for the superiority of humans in performing the sophisticated cognitive and navigational tasks.

Keywords: Biological Reinforcement Learning; Navigational Memory; Meta-Learning

Citation: Lastname, F.; Lastname, F.; Lastname, F. Title. *Nanomaterials* **2021**, *11*, x. https://doi.org/xx.xxxx/xxxxx

Academic Editor: Firstname Lastname

Received: date
Accepted: date
Published: date

1. Preface

To draw insights on human thinking, a new approach championed by computational neuroscientists, and robot designers has been to impart cognition to robots using tools from computer science, in the hope that they can approximate as much as possible the real human behavior. One such tool is Reinforcement Learning (RL), which in fact finds its genesis in bio-inspired neural networks and the learning theories in humans and animal. So it only make sense for theoretical neuroscience of behavioral psychology to try and capitalize on this tool for understanding and acting on human cognition.

RL is the computational framework that works under the hood to give rise to adaptive and active agents (e.g. robots, humans, animals, software) and it rests on the premise that decision making of these agents can be learned only through their own exploration of their environment and through the reward or punishment feedback they receive from the environment as a result of this exploration (spatial or multi-dimensional exploration).

The learning of a basic RL agent occurs during a training phase composed of many trials called episodes, and will stop when the supervisor (i.e. a more omniscient agent that the RL agent itself, usually a human designer) decides the learning achieved is satisfactory, or more commonly when the cumulative reward received during a single episode instance is in its local or global maxima. During training, and under the influence of rewards and punishments, the agent's RL parameters are automatically tuned, and the only

parameters that the omniscient supervisor can tune ahead of time are called the hyperparameters. These variables are design vectors that specifies the size of RL architectures, as well as a host of other technicalities specific to each RL algorithm. The hyperparameters are the designer's outlet to customize the RL algorithm by altering learning speed, patience on gratification, greed and other learning aspects.

The basic incarnation of an RL agent will have to experience a huge number of training episodes before it is deemed fit for deployment, a fundamental problem referred to as "sample efficiency", and the challenge to make the RL agent resemble humans is to increase this sample efficiency to be in par with that of humans and animals. This is where techniques such as imitation learning, transfer of learning, Meta-Reinforcement Learning, and fast RL algorithms using recurrent neural networks (RNN) such as long short-term memory (LSTM) networks enter into play. The idea of this last one, is that instead of freezing the parameters of RL architectures after it was trained, we allow them to carry a memory for the tasks that are to be performed. These state-of-the-art techniques are built on top of intelligent flexible design architectures, and they have proven to generalize well on novel tasks and environments and to carry that generalization along while in deployment phase (i.e. testing phase AKA probing phase).

In general, an ideal RL agent is ought to be adaptive because the same RL architecture should allow for updates on its architectural parameters so that –at least during its training– it learns from different new environments and situational settings all the time (i.e. they are not rigid). An ideal RL agent is also an active one, because efficient RL agents – as their name implies– should have agency on the environment (i.e. controllability) and sampling capability (i.e. observability) through sensors or observers.

Paper Organization: In this anthology paper, we will first define some basic concepts in RL, starting from the Q-learning based RL algorithm, which we deem essential for the characterization of cognition in humans. We expand on some otherwise common RL terminologies, especially those that will be on interest in our study of spatial cognition and make the link between both whenever possible. This include value functions, the greedy policy, exploration vs exploitation, model-free vs model-based RL and on-policy vs off-policy. Importantly, we will present the notions of Temporal Difference learning, temporal discounting, learning rate and tendency to explore with regard to their neural basis in the brain. Next, we will also look at a couple of widely approved models of RL that explain human cognition specifically ones where reward prediction error in encoded by dopamine neurons, modulation of temporal discounting by serotonin and where value coding in realized in the basal ganglia. We will touch on the special role of the hippocampus and striatum which is part of the striatum, and how the interplay between these two and the prefrontal cortex (PFC) give rise to behaviors that are directly correlated with the RL model especially the temporal discounting factor. This paper is also briefly introducing Bayesian inference as a human-like model of learning, and we argue than any model-free RL that is not inferring a model will act unaware of the environment, and will not be as evocative of human cognition as we hope. Our first robot in a maze was an example of such RL devoid of model, which we argue is more baby-like in its approach of foraging and exploring. But at the end, we will present brain models that actually depart from the more complex RL theory, and employ instead the simple supervised learning or unsupervised learning frameworks to approximate learning that take place in certain regions of the brain. Namely, supervised learning performed by cerebellum for the motor coordination and prediction but also for some newly found role in cognition, and unsupervised learning by the cerebral cortical area. And on that score, we will close the paper by describing the merits of realizing those brain models on software and the insight we get from embedding them in a mobile robot to garner diagnostics on psychopathologies that are context navigation-dependent. We will mention extra considerations and implementation modalities that will help devise a biologically-plausible robot. We argue that these cognitive robotic implementations can be an alternative or at least a companion to clinically invasive techniques, and that they can guide psychology practitioners toward ideas on

treatment of non-normative behaviors such as depression, autism spectrum disorder (ASD), and obsessional compulsive disorder (OCD).

It is worth noting that in this study, we will not make distinction between agents that operate in discrete spaces and continuous spaces, although this characteristic will be entertained in our real-world RL applications especially when it comes to reward distributions in the environment (i.e. sparse vs dense).

2. A Primer on RL towards Cognitive Settings

Reinforcement Learning (RL) is the subset of Machine Learning (ML) that endows an artificial agent with the faculties of decision making and action selection (e.g. motor control). With RL, the agent will learn a policy π that dictates the set of actions A_t through which it will achieve its goal in a stochastic (i.e. non-deterministic) environment.

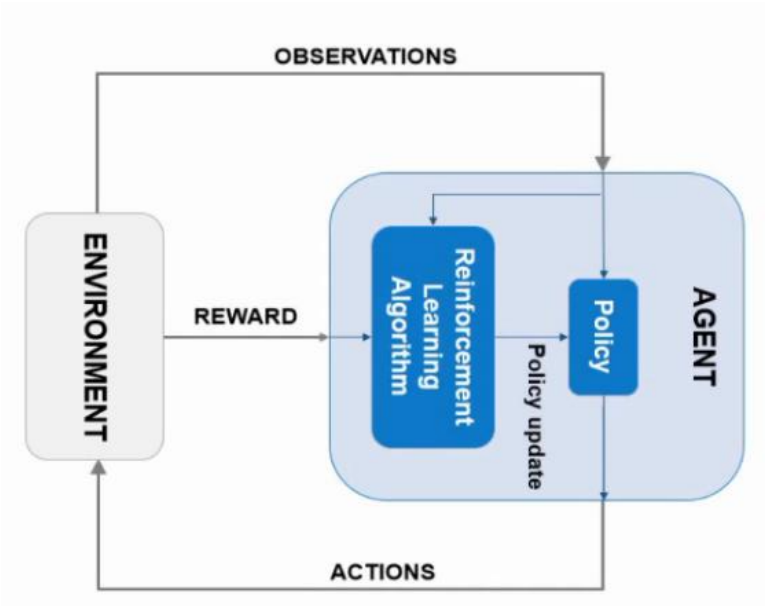


Figure 1. The classical Reinforcement Learning process.

As the flow diagram in Figure 1. is trying to depict, a policy π determines the set of actions A_t, that are taken in observed states S_t (or more precisely, π is said to be a stochastic policy, because it is a probability from 0 ~ 1, that an action A_t is taken in a state S_t). The optimal policy is a policy that accumulate the maximum rewards.

The first step is for the agent to be subjected to a training phase: at each timestep t, it receives observations of its state S (informed either by its sensors or by a state estimator like a Kalman Filter). Based on those observations, it outputs actions A like "turn right" or "move forward". The agent will then see either a reward, no reward or a penalty (i.e. negative reward) which guides it further in training, and hopefully helps it take more rewarding decisions in later steps. This rests upon the optimization of a *value function* V which is the cumulative rewards that the agent collects in each timestep t as it works its way to the goal (i.e. maximizing its rewards). So the optimal value function V is one that yields the maximum value. The question of which actions are contributing a given reward is known as the temporal *credit assignment problem*.

Here is the value function with an infinite time horizon:

$$V(t) = r(t) + r(t+1) + r(t+2) \cdots$$

And as excepted, it is also an infinite sum of rewards into the future. So there is a need to introduce the a discount factor denoted *gamma (γ)* bounded between 0 and 1, so that ***V(t)*** can converge to a finite value:

$$V(t) = r(t) + \gamma^1 \ r(t+1) + \gamma^2 \ r(t+2) + \gamma^3 \ r(t+3) \cdots$$

An intuitive way to see the discount factor *gamma*, is that it makes rewards further in the future worth less than those that will occur earlier. This is very consistent with the behavioral domain in humans and animals. This factor can serve as a tuning parameter to represent different foraging strategies and possibly navigation strategies.

Optionally, we can express this function without an infinity term by writing it recursively as:

$$V(t) = r(t) + \gamma \ (r(t+1) + \gamma^1 \ r(t+2) + \gamma^2 \ r(t+3) \cdots)$$

By substitution of the right-most bracketed term with its equivalent in V, we obtain:

$$V(t) = r(t) + \gamma \ V(t+1) \qquad \textit{Equation} \ (1)$$

The *Equation* (1) teaches us that the goodness of an action taken now can be gauged as the sum of the reward obtained now $r(t)$, and the discounted goodness of the action expected to be taken at the next state $t+1$. This recursive relationship enables an agent to update its belief about the action at runtime (an online update) without having to wait for the protracted rewards of the future.

2.1. Exploration vs Exploitation for psychological behavior

A classical optimization problem that is often encountered in RL is *the Multi-Armed Bandit*, which exemplify the tradeoff between *exploration vs exploitation*. In simple formulation: at each timestep, inferring from prior experience, the agent has to choose between the arm/route that seems to offer the highest expected rewards (i.e. exploitation) and an arm/route we want more data about in hope it might be better (i.e. exploration). The balance of these two strategies is a fundamental dilemma in artificial RL, and crucially to our research objectives, this balance also characterizes a behavior that is observable in humans. Specifically, computational neuroscientists often extrapolate on this RL property to draw conclusions for their psychological studies on humans. (Dezfouli et al. 2019, 2020) have labeled psychological disorders of a sample of population by subjecting them to the Multi-Armed Bandit task and modeling the probability of them switching the left or right arm. This diagnosis characterization was possible by using several models based on Q-learning, Q-learning with perseverance (i.e. alternation), General Linearized Model, and finally they went for a model built around RNN (LSTM) (i.e. non-linearized) which turned to capture better behavioral complexity and yield better diagnostic prediction. Apart from just learning about an existing psychological condition, the authors also used Multi-Armed Bandit tasks to alter the parameters of the models they previously derived in order to specify rewards distribution that can push some aspects of the psychological behavior of subjects and exploit the models maximally.

2.2. The greedy policy emulates curiosity

One important notion tied to this dilemma is that of the *greedy policy*, which refers to an agent constantly performing the action that is believed to return the highest expected reward. Such a policy will not permit the agent to explore at all. As a result, it will likely be stuck in a local optimum, not taking advantage of potential rewarding changes that may happen in the environment. In the real world, human cognition exhibits a "curiosity" element that elicit them to explore occasionally, and so we can say the greedy policy is not

congruent with how humans approach a task like space navigation for instance. A more plausible policy in the biological realm would be the policy that is called *epsilon-greedy policy* where a probability parameter *epsilon (ε)* introduces randomness to the RL algorithm prompting the agent to make random decisions to either exploit and capitalize on its prior experience or explore and uncover potential rewards.

If we choose *epsilon* to be of probability $\varepsilon = 0$, the agent never explores and only exploit the experience it garnered empirically from its prior knowledge, which leaves us with a purely greedy policy again. Whereas if *epsilon* is $\varepsilon = 1$, this presses the agent to always explore by picking random actions and totally ignoring its past knowledge, which is also not a desirable policy because we are ignoring precious information that will guide the agent to an optimal behavior. Figure 2. is representing this concept in symbolic terms and how it can be implemented in pseudo-code.

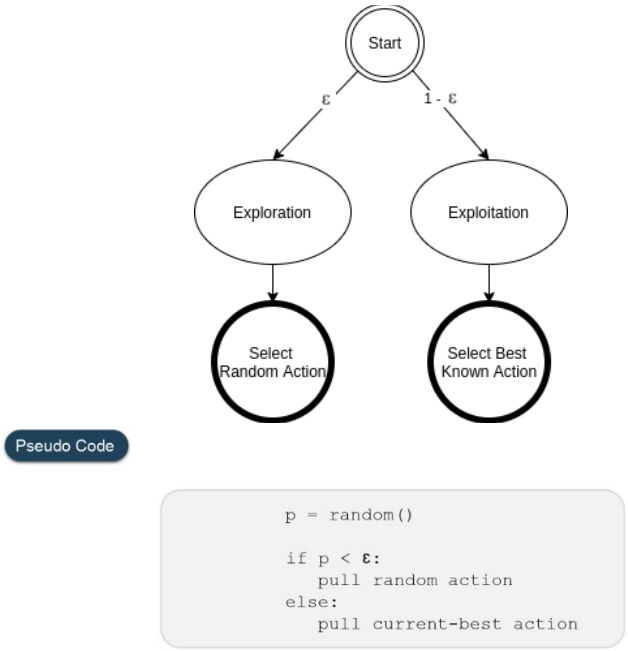


Figure 2. Simplified epsilon-greedy algorithm

To strike a sound balance between both, we should take into consideration the level of knowledge we have of the environment, following a guiding principle that: a totally unknown environment in theory should warrant only exploration, thus an $\varepsilon = 1$, and vice-versa. This balance is aptly illustrated in the chart of Figure 3.

Figure 3. Exploration vs Exploitation in relation to epsilon rate

But in principle, it is recommended to pick *epsilon* ε to be a very small number near zero, so that it largely favors exploitation with occasional exploration so that the agent escapes the sub optimal states. A practical way of going about this, is to start with $\varepsilon = 1$ (i.e. pure exploration), then slowly decaying it by a minute rate with each episode of the training phase, and capping *epsilon* to a minimum near *0*. At the end, we again end up with an *epsilon* ε near zero.

2.3. Q-learning as a basic RL algorithm

One popular application of the epsilon-greedy policy is the *epsilon-greedy Q-learning* algorithm. It is identical to the more ubiquitous *Q-learning* algorithm except when the time comes to select the action: the Q-learning algorithm always selects the action that it believes will yield the biggest reward, without considering to explore other unknown states (i.e. a purely greedy policy for action selection). This makes is very easy to implement, but also shortsighted and less-adaptable for the long run.

Both algorithms however are based on the keeping of a lookup table of *state vs action* pair, which is usually initiated to arbitrary values. Figure 4. is an example of such a table.

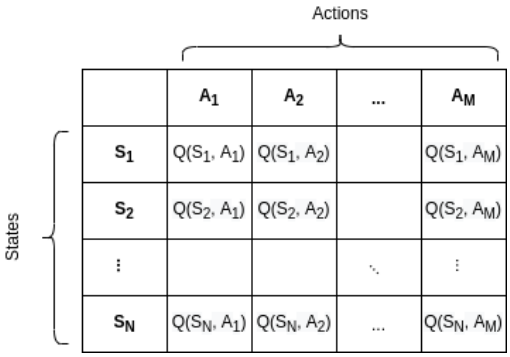


Figure 4. An example of a Q-table with placeholder Q-values

This table, called *Q-table* defines the action A that should be taken by the agent when in state S. The highest *Q-value* $Q(S, A)$ indicate the most optimal action A when in state S.

Note that by the look of this table in Figure 4., one might be under the impression that there is a discrete set of actions and a discrete set of states, while in real life, most environments are made of an infinite number of both. To illustrate this idea, say for example a planar robotic arm with a single joint is using Q-learning table to try to learn the optimal joint angle toque that will get a rod to stand upright, or that will flip a pancake on a pan it holds on its end effector. The actions in this case are measured in term of torque action in *N.m*, and it is obvious that this number is continuous and is only limited by the resolution of the arm's servo motor (or stepper motor thereof) and maximum actuation authority that it can output. The same thing can be said of the state, which in this case is measured in term of degree angle in *radians* of the arm's joint. A table that accommodate these pair of (state; action) will be confusedly large and this will take a huge toll on the memory and computational power during the table update and look-up process. A workaround to this problem, is to discretize the states and action spaces into ranges of states and ranges of actions of admissible sizes and that can be treated as one single observation and one single action selection whenever the actual state observation or action output falls within that particular range. This hand-engineered discretization can either be a naïve discretization (i.e. uniform) or adaptive discretization (i.e. parametric).

Crucially to our spatial navigation exercise, it is true that a mobile robot in a maze can be easily commanded to output clear discrete actions (e.g. turn left, turn right, forward, backward), but when it comes to state, it will see observations that varies widely in a granular fashion with each timestep. So for example, if we take for state observation the

robot's position, then our Q-table will have to deal with as many states as there are pairs of coordinates in our visitable environment. Moreover, the number of these pairs will increase exponentially the bigger is our maze arena or the more is the resolution of our coordinates observer (e.g. GPS or odometry). As hinted before, the solution lay in discretizing the space, in this case, a tessellation of the maze, in a grid fashion or checker fashion will make sense. This will reduce the number of coordinates to a manageable finite number, because the robot will report only on the identifier of the tile it sits on within the maze, and that would be enough for position state.

In general, Q-learning is not advisable for application with continuous states or actions. Its augmented rendition, the Deep Q-Networks (DQN) is more suitable for these dense spaces. But even then, RL design recommendations stipulate that states and actions that bloat in their dimension will slow the RL and since computational operation is usually pipelined, this bottleneck will slow down other RL components down the line. There is also the fear of overfitting the deep networks when too much non-essential state and action data is taken in consideration, when otherwise the network could do without it. This whole issue is a critical one in RL and machine learning in general, and it is termed the *curse of dimensionality*.

The Q-table is updated at each timestep $\boldsymbol{t}$, according to the following formula:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t) \right] \qquad \textit{Equation (2)}$$

The formula of $Q(S_t, A_t)$ is called the *Q-function*. It estimates the sum of future discounted rewards, where R_{t+1} is the reward observed after action A_t has been taken, and *alpha* (α) and *gamma* (γ) are hyperparameters controlling the learning rate (i.e. step-size) and discount factor respectively.

The following pseudo-code in Algorithm 1. and Algorithm 2., concisely describes the process of updating the Q-table using the epsilon-greedy Q-learning algorithm:

Algorithm 1: Epsilon-Greedy Q-Learning Algorithm

```
Data: α: learning rate, γ: discount factor, ϵ: a small number
Result: A Q-table containing Q(S,A) pairs defining estimated
        optimal policy π*
/* Initialization                                          */
Initialize Q(s,a) arbitrarily, except Q(terminal,.);
Q(terminal,.) ← 0;
/* For each step in each episode, we calculate the
   Q-value and update the Q-table                          */
for each episode do
    /* Initialize state S, usually by resetting the
       environment                                         */
    Initialize state S;
    for each step in episode do
        do
            /* Choose action A from S using epsilon-greedy
               policy derived from Q                       */
            A ← SELECT-ACTION(Q, S, ϵ);
            Take action A, then observe reward R and next state S';
            Q(S, A) ← Q(S, A) + α [ R + γ max_a Q(S', a) - Q(S, A)];
            S ← S';
        while S is not terminal;
    end
end
```

Algorithm 2: Epsilon-Greedy Action Selection

Data: Q: Q-table generated so far, : a small number, S: current state
Result: Selected action
Function *SELECT-ACTION(Q, S, ϵ)* **is**
    n ← uniform random number between 0 and 1;
    if $n < \epsilon$ **then**
        A ← random action from the action space;
    else
        A ← maxQ(S,.);
    end
    return selected action A;
end

3. The Brain as a Parallel Processor

This is the right time to remind the reader that the brain is a massively parallel information processing system. And so if this two Q-learning pseudo-algorithms were to run in the region of the brain in charge of cognitive learning, it would certainly not run sequentially from top to bottom like it is the case in general-purpose computers. Rather, all (or some) of the lines of instruction would be executed concurrently (i.e. all at once), like actual neurons in the brain do. Indeed, it has been proven time and again that the neurons in our nervous system are always doing work in parallel. Natural neurons are not clocked in a serial pipeline like it is the case for the "instruction decoder" of CPUs. So, if the intent of the computational neuroscientists is to synthesize a true-to-life model of a biological nervous system, then doing that on a general-purpose computer using traditional sequential programming languages will not be ideal. A more biologically accurate way is to implement their models on FPGAs, GPUs and custom designed CPUs. The silicon logics in these hardware platforms are laid out in a parallel distribution, and so they inherently support parallel programming languages like Verilog, VHDL, SystemC and CUDA API (from Nvidia). These languages –although geared toward graphics and system-on-chip design– can serve well to describe the topology and signal dynamics of brain models.

A lot of work has been done in creating parallel computers that are only dedicated to brain modelling. These computers come in different hardware configurations and memory sizes, but the common denominator between them is that they all use a standard neuron model called spiking neural network (SNN) as their building block. SNN are highly parallelizable, reconfigurable and widely accepted in computational neuroscience. To take advantage of these hardware platforms, the users first have to isolate the natural neural networks of the brain that is of interest to them, then turning them into computer model using the provided software stack that make the platform accessible. And because the model is just a piece of software running on a computer, if the user wants to test a different hypothesis by slightly altering the model, then the platform can almost certainly support that. So these dedicated platforms offer flexibility in neural models, flexibility in the synapse models, and in learning rules, since all these abstractions are implemented in software. What is implemented in hardware is the way the spikes are delivered (i.e. SNN).

The overarching principle here, is that for a maximum realism of the brain learning models, our candidate RL algorithms (including those based on artificial neural networks) should not be bounded by the constraints of the Von-Neumann computer architecture. It is this computer architecture that is dominant in consumer market computers of today, and which prescribes that a computer instruction is to be fetched-decoded-executed one at a time, and to be synchronized by a clock cycle. But as it has been proven, the natural neural networks are not pipelined, not even clocked (i.e. they are asynchronous). And although modern computers have provisions for multithreaded execution, and for SIMD (Single Instruction/Multiple Data) and MISD (Multiple Instruction/Multiple Data), this

does not dispense from the fact that the brain models will be constrained still by different bottlenecks of data and addressing lines.

4. An Account about Neuromodulators in RL Formulation

Following, we will break-up the Q-function to its key defining pieces, and show that each piece corresponds to a particular neurotransmitter in the brain, as evidenced by the seminal work of (Doya 2002: Metalearning and neuromodulation) and the subsequent psychology findings and theoretical models that ensued

4.1. The *Dopamine* as Temporal Difference error

The expression $R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)$ is called the *Temporal Difference error* (TD error and denoted δ). At each timestep we advance into the future, it measures how the last action is different from the action estimated initially in terms of reward. This makes Q-learning qualify as a Temporal Difference learning algorithm, in the sense that the value function V is partially updated online without waiting for the final reward. The TD error expression is the main learning signal in RL algorithms like Q-learning. This expression is also a form of *Bellman equation* in the context of Q-learning, as both of them are optimization problems solvable through a technique pioneered by Richard Bellman known as *Dynamic Programming*. In Dynamic Problem, instead of solving a complex problem in one take, we break it into a sequence of simple sub-problems, as Bellman's "principle of optimality" recommends. Then for each sub-problem, we compute and store the solution. If the same sub-problem is encountered again, we do not need to recompute it, instead, we use the already computed solution.

It has been well established by (Doya 2002) and (Schultz, W., Dayan, P. & Montague, 1997) that there are parallels between the phasic signals emitted by midbrain dopaminergic neurons and Temporal Difference RL algorithms in general. It is therefore said that TD error δ is representative of the *Dopamine* neurotransmitter (DA). Dopamine is an organic chemical that is both a hormone and a neurotransmitter. Contrary to the popular belief, dopamine is not secreted in response to pleasure, but rather in the pursuit of pleasure: namely, it shoots up in the brain in anticipation of a reward, and then quickly returns to normal level once the reward (i.e. pleasure) is attained. Moreover, Schultz and colleagues demonstrated that the tonic firing of dopamine neurons surge to its highest levels when the forthcoming delivery of reward is uncertain, at 50% (Fiorillo, CD, Tobler, PN, and Schultz, W (2003). "Discrete coding of reward probability and uncertainty by dopamine neurons."), which provides an explanation for the special hook that gambling and betting games can have on certain people.

More generally, dopamine encodes the difference between the reward obtained and the reward expected (Schultz, Dayan, and Montague 1997). This difference can be either positive, null, or negative, given these three situations:

① Positive: When zero reward is expected, yet it is followed by positive reward: in this case, dopamine is released at the reward obtention.

② Null: When a reward is expected, and indeed is followed by a reward: The reward is expected due to some stimulus (visual cue, auditive tone, etc..). In that case, the dopamine is no longer released at the reward obtention, but at the onset of the stimulus itself.

③ Negative: When a reward is expected, but it is not followed by a reward: The brain is "disappointed", and reflects that by an acute dip in the dopamine level (i.e. a tonic reversal). This posology is reflecting a negative reward prediction error.

This is exactly how TD error signal is defined in computer RL algorithms. And it is because of this agreement between the dopamine neural response and the TD signal and other similitudes (Waelti et al., 2001; Satoh et al., 2003; Nakahara et al., 2004; Morris et al., 2006) that the literature is relatively confident about the idea that the striatum –being the major target of midbrain dopamine neurons– is the locus of a TD-type Reinforcement Learning (Barto, 1995; Houk et al., 1995; Montague et al., 1996; Schultz et al., 1997). We

must also note in that regards, that the striatum is the brain region that is most responsible for addictive and OCD-like behaviors {need citation}, which further corroborate this confidence.

In this vein, Figure 5., Doya 2000 proposes a comprehensive model that projects the value function signal $V(s) = Expected\ \ [V(t) \mid S(t) = S]$ from its believed origin in the striatum to the midbrain dopaminergic neurons which in turn are broadcasting back to it a TD error signal just like in a RL algorithm. This widely accepted model starts from the premise that the basal ganglia (and the striatum by extension) is the origin of RL in the brain.

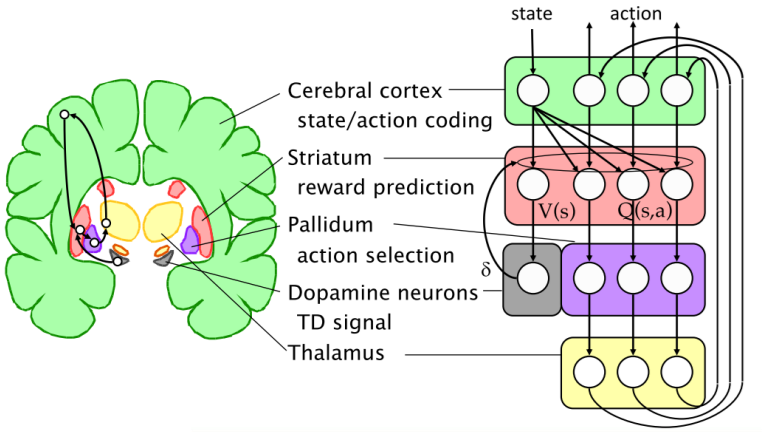


Figure 5. A model-based diagram of RL coupled with the basal ganglia

That being said, the models where the basal ganglia is the basis of RL in the brain are not unanimous. RL-type of learning can be grounded in other brain structures. A study by (Balleine and Killcross, 2006) has shown that the amygdala and the limbic system are also responsible for learning from reward and punishment. TD learning has also been observed in cortical areas, such as the orbitofrontal cortex (Schultz et al., 2000), the prefrontal cortex (PFC) (Watanabe, 1996; Matsumoto et al., 2003, 10.1073/pnas.0500899102 2005), and the parietal cortex (Platt and Glimcher, 1999; Dorris and Glimcher, 2004; Sugrue et al., 2004).

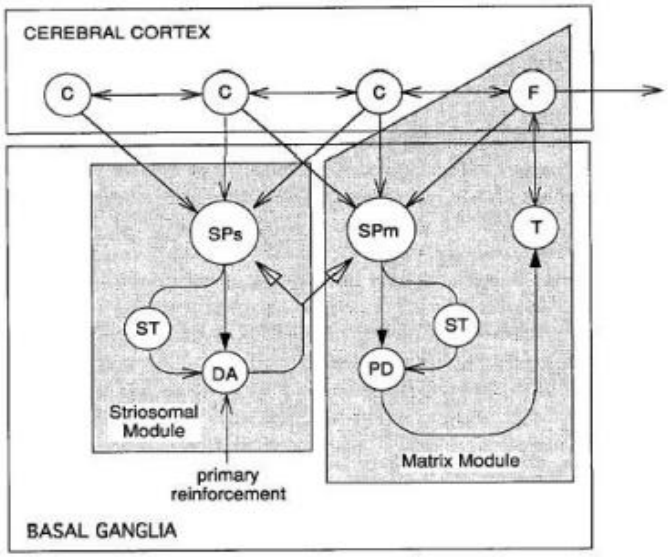


Figure 6. Actor-critic-based model of basal ganglia supporting RL. *Nomenclature*: Open arrowheads signify net excitation (16 arrows); black-filled arrowheads signifies net inhibition (3 arrows) and outlines arrowhead signifies neuromodulation (2 arrows). *Abbreviations*: C: cerebral cortical columns; F: columns in frontal cortex; ST: subthalamic sideloop; SPs: spiny neurons in striosomal

compartments of the striatum; SPm: spiny neurons in the matric compartments of the striatum; DA: dopamine neurons; T: thalamic neurons

There have been other proposed models that attempt to provide plausible pathways for TD error signal in the brain neural network (Houk et al., 1995; Montague et al., 1996; Suri and Schultz, 1998; Contreras-Vidal and Schultz, 1999; Doya, 1999, 2000; Daw and Doya, 2006). They all agree that the TD error signal is mediated by dopamine and that it is used to learn action selection. Houk, JC, Adams, JL, and Barto, AG (1995). "A model of how the basal ganglia generate and use neural signals that predict reinforcement." in particular, poses this problem as an actor-critic model as the architecture in Figure 6. outlines.

The build an intuition on the cognitive significance of the TD error δ, we trace back its origin to *Equation* (1) of value function $V(t)$. If we subtract $V(t)$ from both sides of *Equation* (1), we get:

$$0 = \big(r(t) + \gamma \ \ V(t+1)\big) - V(t)$$

The zero of the left-hand side indicates that the reward is expected and indeed delivered (i.e. situation ②). But when the agent is thrown off by the outcome of its expectation (i.e. situation ① and ③), the term on the left-hand side will not be trivial, and in fact the extent to which it is not zero is the extent to which there is a reward prediction error. Therefore, the term in the left-hand side is nothing other than δ:

$$\delta = \big(r(t) + \gamma \ \ V(t+1)\big) - V(t) \qquad \textit{Equation} \ (3)$$

Equation (3) already has the same formulation layout as the TD error expression. If we substitute the value function $V(t)$ with its homologous Q-function $Q(S, A)$, recalling that the two functions accumulate the same reward, with the exception that Q-function is just expanding on $V(t)$ (i.e. tracking S and A rather than timestep t), then we will deduce the same TD error expression as it appears in *Equation* (2).

If TD error is really the RL parameter for dopamine, and since TD error is just this uniform scalar value, then this raises the question: How can such featureless feedback maps to dopaminergic activity with its rich effects on the brain? this question was answered by (Computational models of reinforcement learning: the role of dopamine as a reward signal R. D. Samson • M. J. Frank • Jean-Marc Fellous 2010) by presenting neurocomputational models of RL that demonstrate a harmonious synergy between brain areas and explaining how they can be implemented by the nervous system, from synaptic to system level.

4.2. The *Acetylcholine* for learning rate

Alpha (α), the learning rate is a hyperparameter encountered throughout the field of machine learning. And as it can be seen from *Equation* (2), the new Q-value for a new state is calculated by incrementing the old Q-value by an amount equal to *alpha* multiplied by the TD error of the selected action's Q-value.

Alpha is a number between 0 and 1 ($0 \leq \alpha \leq 1$). If we set *alpha* to zero, the agent learns nothing from new actions. On the other hand, if we set *alpha* to 1, the agent completely ignores prior knowledge and only reckons the most recent information. Greater *alpha* values make Q-values change faster.

Alpha emulates the function of the neurotransmitter *Acetylcholine* (Doya 2002), which is the chief chemical signal that controls parasympathetic nervous system, itself responsible for regulating muscles contractions, sweating and blood vessels dilatation, but this motor role it sustains also mediates a cognitive role.

4.3. The *Serotonin* for temporal discounting factor

Gamma (γ), the discount factor (AKA temporal discount), is the hyperparameter that scales the estimation of the "optimal future value" (hence the $\max_a$ function), and thus it ascribes a value for the importance for any next rewards. Greater *gamma* values lead to great Q-values.

Gamma according to (Doya 2002) emulates the function of the neurotransmitter *Serotonin*. This molecular hormone plays a wide range of functions the bulk of which is outside the brain like adjusting blood pressure and gut movement, as well as in the brain, like modulating mood, appetite, cognition, learning and memory. And most relevant to us, it has a key role in either eliciting or refraining impulsive behavior (M.J. Crockett, L. Clark, T.W. Robbins 2009)(S. Xu, G. Das, E. Hueske, S. Tonegawa 2017), and either expediting or protracting gratification and rewards (Winstanley et al., 2006)(K.W. Miyazaki, K. Miyazaki, K. Doya 2012)(K.W. Miyazaki, K. Miyazaki, K.F. Tanaka, A. Yamanaka, A. Takahashi, S. Tabuchi, K. Doya 2014). A shorthand definition of serotonin is that it is an opponent of dopamine (Daw et al., 2002)(Y.L. Boureau, P. Dayan 2011), in the sense that it encodes our expected punishment.

Gamma as an RL hyperparameter, is set between 0 and 1 ($0 \leq \gamma \leq 1$). If it is zero, the agent completely ignores the future rewards, and values only the current immediate reward (i.e. it has no patience for gratification). Conversely, if it is 1, the agent would seek for high rewards in the long term (i.e. tolerate delayed gratification). Crucially, this behavior is exactly what psychologists and economists call "intertemporal choice", which in humans is very detrimental in their decision making. Depending on which extreme is their *Gamma* hyperparameter set at: zero or one, the individual is respectively either exhibiting rushing impulsiveness or self-control.

In connection to this interplay, an important discovery by (Tanaka D, Aoki R, Suzuki S, Takeda M, Nakahara K, Jimura K (2020) Self-controlled choice arises from dynamic prefrontal signals that enable future anticipation) has found that these two opposite behaviors resemble a tug-of-war of sort between the striatum, and the frontal pole of the prefrontal cortex (PFC). The former promotes impulsiveness and reward shortsightedness while the latter reflects the anticipatory merit of waiting for a reward even when it has never been experienced before. The participants in this experiment were promised a large dose of juice in an untold future if they could abstain from the temptation of reaching to the smaller dose of juice that is immediately available. Those who controlled themselves when tempted by the immediate reward were doing so even though they have never experienced the taste of the delayed reward, nor told how long they would have to wait for it. The fMRI (Functional Magnetic Resonance Imagery) done during this experiment, reveals that if a participant chose the immediate reward and if their striatum became more active once they started consuming it, then their subsequent choices of when to take the next reward became overtly more impulsive. The fMRI also shows that PFC activation was greater in self-controlled individuals, and that signals coming from this region of cortex will be successful in inhibiting their ventral striatum. This study concludes that greater prefrontal activity contributes to forming strong self-controlled choice preference, which potentially lead people to behave optimally by attaining maximal rewards in the long term. So, we can surmise that greater *gamma* reflects greater activity in the PFC.

A similar inhibition-activation interplay has been observed by (K. Yoshida, M.R. Drew, M. Mimura, K.F. Tanaka 2019), but this time instead of the ventral striatum, it was the ventral hippocampus (vHC) which was inhibited by the median raphe nucleus (MRN) serotonin neurons. This suppressing activity emanating from MRN serotonin took place while mice were engaged in the exercise of level pressing meant for reward-seeking or punishment-averting. In the exercise of traversing a maze conducted by (A.R. Abela, C.J. Browne, D. Sargin, T.D. Prevot, X.D. Ji, Z. Li, E.K. Lambe, P.J. Fletcher "Median raphe serotonin neurons promote anxiety-like..." 2020) and (Y. Ohmura, K.F. Tanaka, T. Tsunematsu, A. Yamanaka, M. Yoshioka 2014), stimulation of serotonin neurons in the MRN elicited anxiety-like behavior in the mice, but stimulating serotonin neurons in the dorsal

raphe nucleus (DRN) did not have the same effect. In (A.R. Abela, C.J. Browne, D. Sargin, T.D. Prevot, X.D. Ji, Z. Li, E.K. Lambe, P.J. Fletcher "Median raphe serotonin neurons promote anxiety-like..." 2020) it is the dorsal hippocampus (dHC) which saw an increase of serotonin release, while in (Y. Ohmura, K.F. Tanaka, T. Tsunematsu, A. Yamanaka, M. Yoshioka 2014) it is the ventral hippocampus (vHC), with the additional remark that no release of serotonin was noticed in the dorsal striatum in that case. One possible reason for this last observation, is that serotonin neurons in MRN heavily diffuse serotonin to the hippocampus, but not to the striatum, whereas those in DRN heavily diffuse to the striatum and mildly to the hippocampus.

Serotonin remains the most elusive neuromodulator, and as we have seen, it is impossible to confine serotonergic neuromodulation to only one generic type of message or role within the brain. A seminal research by (https://www.sciencedirect.com/science/article/pii/S2352154621000255) delved into this point, and has unveiled two important revelations about serotonin: (1) that serotonin neurons projects different messages depending on which target (i.e. brain area) it is talking to. The authors postulate that serotonergic neurons in different nuclei or even within the same nucleus, have somehow self-evolved to fine-tune their messages and effects to better cater for the target recipients. (2) that serotonin signals encode the availability of time and resources required for making decisions as to what action to take. The authors have put together a table (Table 1.) that delineates the different behaviors of the agent depending on the amount of time it has available, and how this tie to serotonin neuromodulation:

{insert table of https://www.sciencedirect.com/science/article/pii/S2352154621000255}

This important tabulated review, in addition to the literature we advanced earlier about serotonin, proves that serotonin (and the *gamma* temporal discount by the same) have the most salient effects on the agent, and that several behaviors emerge as a result of the serotonin-encoded available time.

Looking from an RL-wise perspective, the available time is the factor that promote long-term prediction, extended exploration, and a more tapered learning curve overall. The opposite happen when little time is available time, which may prompt the agent to make uncalculated choices and pressure it lo learn fast but insufficiently.

We can totally leverage on this strong attribute of serotonin when building a model of the brain sectors that cater for navigation strategies. This will mean making granular changes on the *gamma* pertaining to the RL algorithm that power our agent, and then relate the exhibited agent behaviors to the hypotheses that we want to test and validate.

In addition to this explicit tuning of the corresponding serotonergic hyperparameter in the agent, we can also have an implicit way of tuning the *gamma* to achieve similar effects, and that is by changing ambient light: serotonin level in the brain is directly affected by light, where it has been observed that its level increase during day and in summer, and plummets during night and in winter (S.N. Young, How to increase serotonin in the human brain without drugs (2007). Given that serotonin play the major role in brightening the mood, it is not surprising that light therapies are common treatment for seasonal depressive symptoms, and rehabilitating people who experienced low-light confinement (E.C. Azmitia, Evolution of serotonin: sunlight to suicide (2020)). This implicit altering of *gamma* can be realized by incorporating light receptors to our model that will scale *gamma* accordingly. This will make meaningful difference between an agent subjected to the navigation exercise in a dim-lighted environment or one that enjoy it in a well-lit environment, and if this light source is pulsating, static, traveling or point-localized. And because during navigation, individuals do not choose their strategies deliberately, it could turn that the lighting setting has an unsuspected and inconspicuous role in influencing one navigation strategy over the other.

4.4. The *Noradrenaline* for exploration

A fourth neurotransmitter that has also been discussed by (Doya 2002) is the *Noradrenaline* (AKA Norepinephrine) which in reduced levels is known to cause lack of motivation, low blood rate and a faint energy. (Doya 2002) associated Noradrenaline with the property of exploration drive in humans and animals. We see that this property is cognate with the tendency for some humans to always seek the safest route, while other will seek an audacious route where they can garner the most rewards, as alluded to by the "cliff walking task" in (Sutton & Barto, 1998). While this neurotransmitter is not directly represented by any term from within the Q-learning update formula itself, we believe however that this "adventurous" property and its ascribed neurotransmitter can be emulated by the *epsilon* parameter in the case of epsilon-greedy Q-learning algorithm.

4.5. Interim conclusion

To recapitulate the learning role of these four neuromodulators: the dopamine informs the error in reward prediction, the acetylcholine tunes the speed of memory update, the serotonin tunes the time scale of reward prediction, and noradrenaline tunes the randomness in action selection (i.e. noise parameter in stochastic action exploration).

Supplementary information about the role of these neuromodulators are still coming. For example, (Yu and Dayan, 2005) have constructed a model which affirms that acetylcholine encodes the expected uncertainty, (e.g. when an agent starts navigating in an unvisited environment), and that conversely, noradrenaline encodes the unexpected uncertainty (e.g. when the learned environment suddenly changes). Their model enjoys a strong validation, since it successfully justifies the results of pharmacological experiments in attention tasks.

5. On-Policy and Off-Policy RL

In the *Equation* (2) of the Q-function, notice that the action small a refer to the action that is taken assuming a greedy policy, even if we are talking here about the Q-function of the epsilon-greedy Q-learning algorithm. This means that Q-learning always assumes a greedy policy (i.e. optimal) is followed even though in reality it might not be following a greedy policy. It is for that reason that Q-learning algorithms are called *off-policy* learning algorithms. In off-policy learning algorithms, the (greedy) policy that is evaluated and updated (called "estimation policy" or "target policy") is different from the (epsilon-greedy) policy that is used to select actions A (called "behavior policy").

In contrast, an *on-policy* learning algorithm would be an algorithm where the same (epsilon-greedy) policy that is evaluated and updated is also used to select actions A. Examples for on-policy RL algorithms are: SARSA, TD(λ) and actor-critic based methods like TRPO, PPO, A2C (Advantage Actor Critic) or its antiquated asynchronous variant, the A3C (Mnih et al. 2016).

For instance, the reason that SARSA is on-policy is that it learns action values relative to the policy it follows, while off-policy Q-learning does it relative to the greedy policy. SARSA learning, updates its Q-values using the Q-value of the next state S_{t+1} and the current policy's action A_{t+1} (the behavior policy).

Q-learning: off-policy, because target policy is always greedy policy:
$\delta(t) = R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)$

SARSA: on-policy, because target policy is always same as behavior policy:
$\delta(t) = R_{t+1} + \gamma\, Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)$

Under some common conditions, they both converge to the optimal value function, but at different rates. Q-learning tends to converge a little slower, but has the capability to keep learning action values by serving actions that it did not execute during the learning phase. Another interesting property of off-policy Q-learning algorithm, is that you do

not need to follow any specific policy, your agent may even behave randomly and despite this, off-policy methods in general can still find the optimal policy. This quality was documented in (Watkins 1992).

6. Meta-Reinforcement Learning for general cognitive intelligence

But one of the major shortcomings of all these previous RL agents is that they are unable to generalize the learned policy to newer problems. A previously learned rule would cater only to the specific problem it was trained for, and would often be useless for other (even similar) cases. Humans and other biologically intelligent creatures do not suffer from this policy stiffness, and can adapt to various datasets and tasks even when they are novel to them (i.e. they are task-agnostic agents).

A more biologically plausible variant of RL is the Meta-RL (i.e. Meta-Reinforcement Learning) where the policy π adapts to the task and scenarios met after training, essentially making the agent learn "on the fly" during deployment stage. The formal description of Meta-RL is that: given a *distribution over environments*, it trains a *policy update rule* that can solve new environments given only limited or no initial experience.

In basic RL, the policy optimization is formulated as: given a *distribution over examples* (i.e. a single task), learn a *function* that minimizes the loss:

$$\hat{\phi} = \arg\min_{\phi} \mathbb{E}_{z\sim\mathcal{D}}\left[l\left(f_{\phi}(z)\right)\right]$$

Where $\mathbb{E}_{T\sim P}$ is a distribution over examples.
And $f_{\phi}(z)$ is the loss function (AKA cost or objective function).

A step beyond that is Meta-Learning (i.e. Learning-to-learn) in which we take *a distribution over tasks* and output an *adaptation rule* that can be used at test time to generalize on unseen tasks and datasets. And this is formalized mathematically as follows:

$$\hat{\theta} = \arg\min_{\theta} \mathbb{E}_{T\sim\mathcal{P}}\{\mathcal{L}_T[g_{\theta}(T)]\}$$

Where $\mathbb{E}_{T\sim P}$ is a distribution over tasks or datasets.
And $\mathcal{L}_T[g_{\theta}(T)] \coloneqq \mathbb{E}_{z\sim\mathcal{D}_T}\left[l\left(f_{\phi}(z)\right)\right]$
In which $\mathbb{E}_{z\sim\mathcal{D}_T}$ is a distribution over examples of tasks T.
And $g_{\theta}(T)$ is the adaptation rule that takes a task description T and outputs a model.

And given that Meta-RL understanding is nested upon this Meta-Learning mathematical model, which itself is nested upon simple RL, we can say therefore that Meta-RL is another step beyond regular Meta-Learning and beyond RL per se. It a universal learning paradigm that approaches the actual learning in humans and that can hopefully mimic human cognition. The field of Meta-Learning in general is seen as the state-of-the-art solution to the limitation enunciated by [Rich Sutton, Anna Koop, David Silver (2007)] in which they say that: "Machine learning can be characterized as the search for a solution that, once found, no longer need be changed". Machine learning has been more concerned with the results of learning than the ongoing process of learning, while Meta-RL is lending itself well to both the result of learning and the rejuvenation of learning. (Wang, J. X., Kurth-Nelson, Z., Kumaran, D., Tirumala,D., Soyer, H., Leibo, J. Z., Hassabis, D., and Botvinick, M. Prefrontal cortex as a meta-reinforcement learning system. Nature neuroscience, 21(6):860, 2018.) pioneered the idea that the PFC is the brain region where Meta-Learning is processed suggested by the considerable number of recurrent neural connections that make up the PFC network.

In the context of spatial navigation task, Meta-RL agents have been perfectly successful in exploring mazes and open spaces in a human-like way {fast RL2:

https://arxiv.org/pdf/1611.02779.pdf and https://arxiv.org/pdf/1604.06778.pdf }. The basic idea of Meta-RL in a maze setting can be distilled in the Figure 7. block diagram.

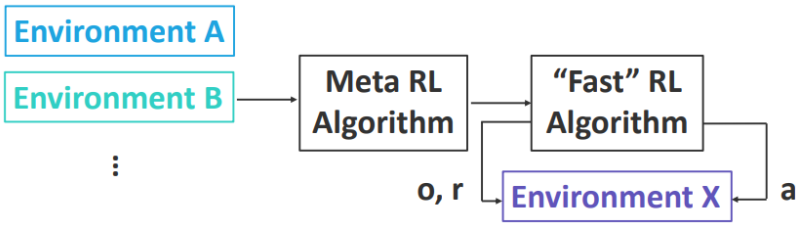


Figure 7. Meta-RL invocation within the problem of maze navigation

7. Model-Free or Model-Based RL

In a model-free Reinforcement Learning, the agent learns only from trying different options and probing the outcome, without constructing an internal model or a *Markov Decision Process* (MDP) [Richard S Sutton and Andrew G Barto. Reinforcement learning: An introduction, volume 1. MITpress Cambridge, 1998.][Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare,Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level controlthrough deep reinforcement learning. Nature, 518(7540):529, 2015.][David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche,Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Masteringthe game of go with deep neural networks and tree search. nature, 529(7587):484, 2016.]. This definition stems from the fact that model-free RL systems are not Markovian, i.e. they do not satisfy Markov property: the probability of the next state given just the current state and current action is equal to the probability of the next state given the entire history of states and actions; formally:

$$P(x_{t+1} \mid x_0, a_0, \ldots, x_t, a_t) = P(x_{t+1} \mid x_t, a_t)$$

And because of this property, a model-free RL is believed to be close to the way a baby is learning, not the way an adult is. That is because a baby assumes no a-priori knowledge of the world and go through many trials-and-errors before he or she adopts a proper policy that gets him or her to the reward. This policy is not necessarily the optimal policy, but very close to it. For example, a baby learning how to walk, might tradeoff energy over accuracy: babies are known for their clumsiness, and how they often fail to stop right where their target is, but just close enough. In other words, babies will concede to mediocre control provided it can saves them precious calories worth of energy. This frugal strategy of babies finds its analogy in control theory: it is equivalent to placing the poles (i.e. eigenvalues) of a dynamic system close to the imaginary axis of the complex plane in order to save on actuator effort, even when this means the response of the system will be slower. *(at 7:20 youtu.be/ClDjWyHRLks he talks about Imitation Learning for child)*

Therefore, in the case of traversing a maze, implementing a model-free RL is more representative of how a baby would approach that task, and less of how an adult would do the same. Indeed, an adult comes already equipped with a physical model of the 3D world and its dynamics. For instance, if adults perceive a wall even from far away, they will immediately deduce it is a dead-end. Or in the presence of an occluded or noisy visual cue, they will be perfectly capable of reconstructing the full cue and thus take heed of it, even if it gets out of their line of sight.

7.1. Bayesian Inference in Partially Observable States

In the real-world applications, the RL agent often do not have full observability of its own state (i.e. partially observable). While this lack of information is trivial for a human, it is a big challenge for an artificial agent, as it will compromise its policy update and thus

its decision making, compared to if it was operating in a fully observable environment. This type of problems is known as *Partially Observable Markov Decision Processes* (POMDP) [Astrom, K. J. Optimal control of Markov processes with incomplete state information. Journal of Mathematical Analysis and Applications, 10(1):174–205, 1965.]. To resolve this uncertainty of state, RL practitioners usually borrow solutions already in use in control theory, like the Kalman Filter with all its augmented versions, the Particle Filter, or the Graph Optimization. These solutions are all predicated on the ability of the agent to fuse the past and current ambiguous observations (i.e. measurement) with an internal model of the environment (i.e. a process) that can predict the state either using past measurements or more generally using a mathematical representation of the environment dynamics. As a result, the agent will then converge on a "belief state" that closely captures its real current state (Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. Artificial intelligence, 101(1-2):99–134, 1998.)(David Ha and Jurgen Schmidhuber. Recurrent world models facilitate policy evolution. In S. Bengio, ¨ H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), Advances in Neural Information Processing Systems 31, pp. 2450–2462. Curran Associates, Inc., 2018.)(Alex X Lee, Anusha Nagabandi, Pieter Abbeel, and Sergey Levine. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. arXiv preprint arXiv:1907.00953, 2019.). These methods are all grouped under the banner of what is known as Bayesian inference, which is a class of probabilistic inference grounded in statistics. And since these methods provide the RL agent with a predictive model of the world, then it can be said that the Bayesian-based RL algorithms are typical examples of model-based RL. A large body of evidence such as presented in (Knill and Pouget, 2004; Doya et al., 2007)(Akihiro Funamizu, Bernd Kuhn, and Kenji Doya. Neural substrate of dynamic Bayesian inference in the cerebral cortex. Nature neuroscience, 19(12):1682, 2016.) support the theory that the brain cortex realizes Bayesian inference.

In one such evidence, (Yoshida and Ishii, 2006) performed fMRI on participants who were tasked with navigating a virtual maze towards a goal but with limited vision. The result of their brain imaging shows that the anterior prefrontal cortex (aPFC) was activated in response to the uncertainty of the current state, while medial prefrontal cortex (mPFC) was activated when there was a strong discrepancy between the sensory observation and prediction. This study hence conclude that these two prefrontal regions of the cortex infer decision making according to the Bayesian model. This conclusion aligns with a more recent study by (https://www.nature.com/articles/s41583-019-0220-7) on Bayesian Reinforcement Learning that occurs between the PFC and basal ganglia: The authors say that midbrain dopamine signals are not only implicated in encoding TD error as traditionally known, but are also responsible for reporting uncertainty about the belief state of the agent in the environment.

The assumption that Bayesian models underlay the work of the brain has helped design certain clinical experiments on mice that would have otherwise been difficult to execute due to incomplete observation of their internal neuromodulation data (K. Miyazaki, K.W. Miyazaki, A. Yamanaka, T. Tokuda, K.F. Tanaka, K. Doya 'Reward probability and timing uncertainty alter the effect of dorsal raphe serotonin neurons on patience' Nat Commun, 9 (2018))(K. Miyazaki, K.W. Miyazaki, G. Sivori, A. Yamanaka, K.F. Tanaka, K. Doya "Serotonergic projections to the orbitofrontal and medial prefrontal cortices differentially modulate waiting for future rewards" Sci Adv, 6 (2020))

7.2. Our attempt at RL-guided agent in a maze

Our earliest RL incarnation that we designed and embodied in our maze foraging robot {see figure from Webots I will put} was based on two general-purpose model-free RLs, namely the actor-critic DDPG algorithm (Lillicrap 2016: original paper co-authored by D. Silver: https://arxiv.org/abs/1509.02971) and the Deep Q Network (DQN) algorithm (Mnih et al., 2015). It can therefore be considered that our agent was governed by a model-

free RL scheme, and that is why our mobile robot was more appropriately simulating how a baby would traverse a maze. This explains why our wheeled mobile robot was naïve and not aware of walls until it rams into them, and why it took convoluted routes even when the path was clear and straight. It is akin to presenting a baby with the task of retrieving a reward at the end of a maze, when the baby has not yet learned how to walk and navigate in environments without bumping into walls.

One problem that dragged our model-free RL implementation was incorporating physical primitives like "distance to wall" and "speed" into the reward shaping equation, which led to a very low "sample efficiency". Even more damning to our RL implementation, is that even though babies are believed to operate according to a model-free RL, they do not suffer from low "sample efficiency", that is, they can learn from just few training episodes. This few-shot learning quality in babies is mainly due to their ability to generalize well on past experiences to new situations (i.e. Transfer of Learning), combined with an advanced hierarchical processing of sensory input, and overall, an affinity to meta-learning. We can remedy to these limitations by incorporating Meta-Learning and adapting some of the fast RL algorithms to our implementation, such as (https://arxiv.org/abs/1806.03335 [Osband, http://iosband.github.io/2018/11/30/Randomized-Prior-Functions.html]) and RL2 which represent RLs as a recurrent neural network (RNN) therefore equipping them with a memory of the task which follow them in both the training and testing episodes (i.e. a form of episodic memory in humans) [https://arxiv.org/pdf/1611.02779.pdf]. A possible solution that defeat these limitations is by using Progressive Neural Networks as proposed by {https://arxiv.org/pdf/1606.04671.pdf}. In {https://arxiv.org/abs/1611.05763}, several compelling solutions using Meta-RL have been discussed, some of which are geared towards neuroscience. Another novel technique developed by OpenAI researchers (https://arxiv.org/abs/1707.01495) and called Hindsight Experience Replay can also alleviate the poor sample efficiency we encountered in our early RL implementation. It replays experience –a technique often used in off-policy RL algorithms like DQN and DDPG– but with goals chosen in hindsight, after the episode has elapsed. HER agents can therefore learn even though they may have never actually hit the desired goal early on (i.e. learning from failure). This method can only be combined with off-policy algorithms since goals are substituted, making the learning update purely off-policy by definition. So HER can be compounded with any off-policy RL algorithm, including DDPG, in which case it is usually denoted as "DDPG + HER". It could also be merged with other state-of-the-art RL methods like the Prioritized Experience Replay, distributional RL, entropy-regularized RL, or reverse curriculum generation, as highlighted in (https://arxiv.org/abs/1802.09464).

7.3. Emergence of RL in babies

There is also substantiated evidences to support the model-based hypothesis in babies, which states that babies brains come pre-programmed since birth with an intuitive physics and psychological model of the world, both of which infer certain beliefs about it. For example, babies from 0 to 4 months old are already able to recognize that an object they have seen previously still exist even if it gets occluded (i.e. object permanence) [Josh Tenenbaum @CCBM2018][Liz Spelke][..]. This hypothesis is easily verifiable from everyday observation when it is known that newborn babies are already able to swim and eat for example, as well as solve a range of complex cognitive problems and tasks without any supervision (i.e. without human-in-the-loop). This suggests that there is a form of model-based framework already embedded in babies' brain that leverage on physical and psychological truths, and on which RL is built on.

So an important determinant of the RL policy is predicated on these two paradigms of human Meta-Learning:

- The *nurture* hypothesis: is a baby only learning by imitating a mentor and other individuals? hence model-free

- The *nature* hypothesis: is a baby hardwired from birth with an innate model of how walking or gait is to be accomplished? That is to say, the baby plays an internal mental simulation before he or she proceeds with an action, hence model-based.

Several researchers have put forward that these two RL policy update rules operate in parallel to learn action selection. (Daw, Niv, & Dayan, 2005; Keramati, Dezfouli, & Piray, 2011; Pezzulo, Rigoli, & Chersi, 2013).

A number of other studies have recently highlighted hat these two paradigms are not mutually exclusive in one's brain, but instead that they are interlocked and that they can coexist and manifest depending on situational or temporal factors, effectively blurring the distinction between model-based and model-free RL [Y. Worbe, S. Palminteri, G. Savulich, N.D. Daw, E. Fernandez-Egea, T.W. Robbins, V. Voon 2016][Han, Doya, Tani 2020 {openreview.net/forum?id=r1lL4a4tDB}][Dayan @4:25 https://www.youtube.com/watch?v=BokS8PpIpc4]. This hybrid concatenation of two systems agrees with [Kimberly L Stachenfeld, Matthew M Botvinick, and Samuel J Gershman. The hippocampus as a predictive map. Nature neuroscience, 20(11):1643, 2017.] who introduced the idea that the hippocampus is capable of capturing a representation of the environment that is expressed in both model-free and model-based RL. These conclusions are a departure from the mainstream view that the brain is triggered to choose one system over the other [Jan Glascher, Nathaniel Daw, Peter Dayan, and John P O'Doherty. States versus rewards: dissociable ¨ neural prediction error signals underlying model-based and model-free reinforcement learning. Neuron, 66(4):585–595, 2010.][Peter Smittenaar, Thomas HB FitzGerald, Vincenzo Romei, Nicholas D Wright, and Raymond J Dolan. Disruption of dorsolateral prefrontal cortex decreases model-based in favor of model-free control in humans. Neuron, 80(4):914–919, 2013.][Sang Wan Lee, Shinsuke Shimojo, and John P ODoherty. Neural computations underlying arbitration between model-based and model-free learning. Neuron, 81(3):687–699, 2014.].

7.4. A possible link between model-free and model-based RL in spatial navigation

What is pertinent to the navigation strategies domain, is that these two systems could well be modulating the allocentric vs egocentric navigation strategies, in the sense that the mimicking a pattern is more consistent with an allocentric strategy while the mental simulation is more what egocentric strategy would dictates. This modulation can be in the form of a probabilistic finite state machine, where the sense of agency (SoA) is the variable that influences one navigation mode over the other.[Ohata & Tani (2020), http://www.brain-ai.jp/wp-content/uploads/2020/10/PDF_20.pdf]. The probabilistic nature of this state machine makes it part of the Markov Decision Process (MDP) family, specifically due to its characteristic stochastic nature of its decisions and quasi-randomness of its outcome.

8. Biologically Plausible Models for Action Selection

In the basic form of RL, the agent just takes an action by referring to its policy, and it will update it when a reward is announced by the TD error signal that it receives. However, when the agent sees its goal change due to an intrinsic factor (e.g. agent loses interest in the past goal or develop interest in a new goal), then it makes sense to assume that the actions that lead to that goal will also change, even if the dynamics of the environment remained unchanged. This is a typical case when the brain will build an internal model that learns the environmental state transition probability rule $(S_{t+1} \mid A_t)$, and will conjure up that model to carry out a mental simulation that will lead to the new goal (Doya, 1999; Kawato, 1999). Basically, the brain introspects itself by asking: if I perform action A_t in current state t, what new state S_{t+1} will I end up in. In this scenario, if the agent can read out the reward of each state it happens to be in, then the agent can reassess the fitness of any hypothetical action it intends to take.

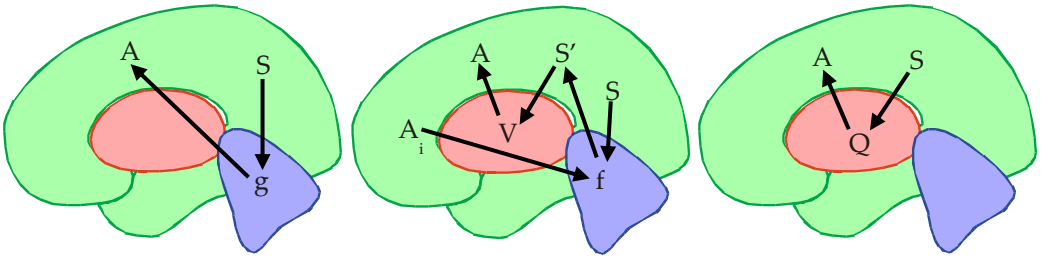


Figure 6. Actor-critic-based model of basal ganglia supporting RL. *Nomenclature*: Open arrowheads signify net excitation (16 arrows); black-filled arrowheads signifies net inhibition (3 arrows) and outlines arrowhead signifies neuromodulation (2 arrows). *Abbreviations*: C: cerebral cortical columns; F: columns in frontal cortex; ST: subthalamic sideloop; SPs: spiny neurons in striosomal compartments of the striatum; SPm: spiny neurons in the matric compartments of the striatum;

Hebbian in cerebral cortex

The work of [Diba & Buzsaki 2007] demonstrates how a form of mental simulation has also been reported in mice before running a maze: electrodes recordings from mice's hippocampi show that a certain cluster of cells were firing at a unique pattern that is directly traceable to their location in a maze. This cluster is the Place Cells. What is equally interesting, is that when the mice are waiting and about to run the same maze again, the exact same firing pattern is recoded again in the Place Cells, but this time it is in backward and in a much accelerated pace. This said neuro-cognitive phenomenon is called *preplay* (the experience will happen again in future). This same phenomenon of reversed-fast sequences of spikes has also been recorded during random times of the mice day, both during wake (i.e. grooming, eating, etc..) and –crucially – during sleep as well. This is referred to as *replay* (the experience has happened in the past, but not necessarily again). This suggests that at least in the rodents' brain, the hippocampus is actively engaged in anticipating the next experiences (i.e. navigation, foraging, motor memorization, etc..), planning its path beforehand, as well as revisiting past experiences.

The phenomenon of replay/preplay is also very reminiscent of the idea of "mirror neurons", which are neurons that fire both when a subject performs an action and again when the subject hears or sees another subject perform a similar action. In the replay/preplay, the "other subject" is the subject itself (i.e. the ego). That is because the firing of the place cells happens not only while exploring the space (as it ought to be), but these cells fire again –albeit in reverse– when the animal is just self-contemplation the action of exploring another time.

The hippocampus has always been identified as the indispensable passage station from which short-term memories (STM) pass through to get consolidated into long-term memories (LTM) before they are subsequently stored in the cortex. This consolidation is possibly happening during what is known as "sleep spindles", which are characterized by REM (rapid eye movement) and can be a catalyzer for dreams during sleep. While this flow is all true, now with the replay/preplay function, we can affirm that the hippocampus is not only responsible for memory (STM & LTM), but also for imagining scenes from the future and from the past (i.e. The hippocampus is our time travelling machine AKA our biological spatiotemporal device). These two seemingly different cognitive functions of memory and imagination are not totally far apart, after all, if we are planning for the future, it usually involves conjuring up fuzzy scenes in our mind's eye.

This imagining of a scene from the past is also akin to "rumination", which in psychology, refers to the recalling of unsatisfactory past events and reimagining alternate scenarios that would have led to a satisfactory output. It is thought that during rumination, the animal is "training a policy" so that when faced with the same task, the subject does not have to go through the expensive computational task of planning all over again. But in humans, this rote hypothetical thinking has been identified as a symptom of acute depression.

Another research (https://www.sciencedirect.com/science/article/pii/S2211124718317960) has been relatively successful in graphically tracing the path trajectory taken by a mouse just by snooping on their hippocampus (i.e. CA1 region) signals using tetrodes (i.e. an array of 36 high density silicon probes) while it explored an open space. Those signal spikes were decoded into point coordinates via GPU so that the tracing can be done online (i.e. in real-time) as opposed to offline (i.e. after the fact). When the software traced path is overlaid with the actual path taken by the mouse, it coincided significantly well. Furthermore, the researchers were also able to see the spiking pattern at an ulterior time, reconfirming the theory that mice are replaying their navigation experience at rest and at sleep either when they dream of running the maze or when the memory of the maze topology and spatial experience is being elevated from STM to LTM.

These two experiments and others involving the neural recording of Place Cells representations have provided evidence that an internal model of the spatial experiences is being built in preparation for a new upcoming round of the same experience, or during sleep when the experience is over, but the brain is suspecting it to occur again in future. This mental simulation is there to imagine best action selection scenarios (i.e. policies) that will maximize rewards along the way, or those that will avert pitfalls from the past. And being tightly coupled with the hippocampus, it is safe to say that this mental simulation is relying on memory and imagination in which the hippocampus is the chief orchestrator.

8.1 Other candidate models of cognitive behavior beside RL

Here we should mention that other models outside the RL realm have been proposed to explain brain learning. One of these is the *supervised learning*, the most basic machine learning paradigm where the training is guided by labeled targets that will help the agent (i.e. brain) learn the mapping function from the input to the output. Another is *unsupervised learning* where the agent is not given any explicit labels or input-output pair examples, yet it is expected from it to autonomously figure out distinct patterns in the input and cast them at the output. RL is seen as a middle-ground between the two supervised and unsupervised learning types: RL learn from labelled targets (i.e. reward) but is not explicitly given the policy from state to action that it should follow, and it should derive it on its own.

With this in mind, some studies have postulated that these two paradigms occur in the brain too alongside RL, with each paradigm being executed by a brain region: the cerebellum is specialized for supervised learning, the basal ganglia for RL, while the cerebral cortex is devoted to unsupervised learning (Houk and Wise, 1995; Doya, 1999, 2000). {include the 3 block diagrams figure in Doya slides}

9. The Case for Software Modeling of Brain's Neural System vs State of the Art

In the same spirit as the many studies we cited throughout this paper, we are also envisioning a software model that will capture how the brain learns, and we want to pay special interest on how it executes spatial navigation tasks. This is a step towards a universal and biologically-plausible model of the brain's circuitry responsible for spatial representation and navigation. By synthesizing our model on software, we can intervene directly in the signals projected in order to mediate different hormonal changes, and then witness the behavioral shifts instantly reflected on the robot in the 3D virtual world.

Traditionally in the neuroscience practice, to introduce an input in target neurons cells, the animal is subjected to pharmacological manipulation or electric stimulation through micro electrodes, however imprecise these methods can be. Today, the more in-vogue method for influencing brain signals online is by mean of optogenetics: the animal scalp is exposed, and neurons are flashed with light to either inhibit or activate those signal channels as the experiment is conducted. Prior to that operation, the animal has to either be infected by light-sensitive viruses, or to be from a breed of genetically modified animal (O. Yizhar, L.E. Fenno, T.J. Davidson, M. Mogri, K. Deisseroth 2011). It is the ability

to switch neurons on and off in living moving animals that make this technique so powerful. Clinical experiments will usually combine both optogenetic infusion and electrical stimulation in what is known as "photo-tagging".

Similarly, this software closed-loop approach we are advocating for, can be combined to the clinical approach (i.e. optogenetics, electrical stimulation, molecular manipulation). Or else, it can even be presented as a standalone alternative to clinical methods in trivial experiments where non-invasiveness, low-cost and scalability are desirable. We believe this workflow will open the door for ground-breaking advances in diagnosing (i.e. prognostic) and treating psychological impairments that find their expression in spatial navigation.

9.1. The cognitive robot as a tool to study OCD-dependent navigation strategies

One of these impairments we are particularly focusing on is that of OCD and its most visible manifestation, perfectionism. One research we are inspired by is that of (https://scholar.google.com/scholar?cluster=8649631303721026231&hl=en&as_sdt=0,5) which invited a population of individuals to explore a virtual maze, and made note of (1) their profile information: age, gender and (2) their performative data:, whether their final navigation trial (i.e. probe) was allocentric (i.e. place strategy) or egocentric (i.e. response strategy), as well as their trajectories metrics collected during this exploration. Then the authors trained a machine learning network for a portion of the population using their data as vector input to the network, and their individual perfectionism scores (which they submitted on a survey port-experiment) as the correct labels of the output (i.e. supervised learning procedure). When put to the test, it turned out that this trained network was successful in predicting the correct perfectionism score probability when injected with data of the remaining portion of the population it was not trained on.

We are certain that matching this data-to-perfectionism score correspondence can well be reproduced in a mobile cognitive robot, but instead of the profile information as one of the classifier network input, we will input the RL agent's hyperparameters, in addition to the performative data that we would have measured after the agent is done with its maze navigation (all the trials and probe episodes).

The software model that runs the robot will be tuned and tweaked until we get a clear correspondence, and from there we can use this model to fulfills our need of psychopathology analysis, prediction, diagnosis and what is more, the online editing of the model may impart us with hints about treatment to these psychological ailments.

9.2. Consideration in implementing the software models in foraging robots

In our baseline task of a robot agent solving a maze using human-like cognitive capabilities, it is important to note that the agent does not enjoy a top-down view (i.e. bird's eye view) of the entirety of the maze in-prior (i.e. agent is not omniscient about the topology of the maze). If the artificial agent had a full map of the maze ahead of time, then it could have easily used any pathfinding algorithm like the Dijkstra, A*, Depth first search, Breadth first search (i.e. search-based algorithms) or Probabilistic roadmap (PRM), Rapidly-exploring random tree (RRT) (i.e. sampling-based algorithms) to trace a path from the start directly to the goal at the end of the maze. But in reality, the humans and rodents can only acquire a partial knowledge of the maze topology, given their limited sensors: they are only informed by their own first-person view through their biological camera (i.e. eyes and cortical visual areas).

A caveat to this seemingly first-person only perspective of humans, is that –at least for rodents– it has been found by (O'Keefe https://doi.org/10.1016/0014-4886(76)90055-8)(https://ui.adsabs.harvard.edu/abs/2005Natur.436..801H/abstract)(https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4315928/) that mice possess special clusters of neurons that endow them with some form of rudimentary geolocalization in space. This effectively provide them with a "cognitive map" of the space that is stored

and rendered in their minds as they explore. These cell clusters are conveniently dubbed: Place Cells, Grid Cells, Boundary Cells and Head Direction Cells, and can be found in the hippocampus and entorhinal cortex. The presence of such cells in humans is very elusive, and cannot be ascertained due to ethical considerations, since neurophysiologists cannot simply plug invasive probes deep into the limbic system of human beings and ask them to perform walking navigational task in the way they can do to rodents. However, some researchers were able to bypass this logistical problem by designing 3D virtual reality navigation tasks that subjects individuals could explore while remaining stationary. The result of their investigations seem to suggest the existence of cells exhibiting place-dependent and grid-like spiking patterns in the human hippocampal formation (https://pubmed.ncbi.nlm.nih.gov/12968182/)(https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3173857/)(https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3767317/). And while it is true that these investigations still required depth electrodes to be surgically implanted in those handful individuals, it must be noted that these individuals were already undergoing epilepsy treatment, and as such these electrodes where there as part of their clinically-mandated epilepsy monitoring, and this an important distinction to make in these type of invasive brain recordings.

But in the event we want to emulate the effect of these context map-dependent cells in our agent robot, we might want to incorporate SLAM and online path planning techniques to draw a cognitive map in the agent memory.

On a side note that has its importance in implementation: it is because Place Cells are located in the hippocampus, that we attribute allocentric navigational strategies to hippocampus. Since after all, one need a learned map of an environment in order to tell location of two objects with respect to each other. It is also useful to recall that another name for allocentric strategy is place strategy.

On the other hand, it is commonly believed that striatum is the brain structure that handles egocentric navigation strategies given how striatum promotes impulsiveness and heedless reward seeking. This striatum influence would inhibit an otherwise more analytical thought process, one which would for example decide that stationary objects in a maze arm do not wishfully move to match our changing starting position, and that it is we who have to recoup for that new stating position by taking a mirrored path.

10. Conclusion and Open Questions

The review paper tried to scope and synthesize the major works of the literature that utilizes RL frameworks for assimilating cognition in agents. Still it remains a brief tour of these extensive ideas that borrow from RL to build insight on the rules that govern cognitive behaviors in human and animals. We started by an overview on the fundamentals of RL theory and its key concepts and classifications, by focusing on the most idiomatic RL algorithm, that is Q-learning. From there, we presented the widely accepted hypothesis about how neuromodulators, such as adrenaline, serotonin, acetylcholine and noradrenaline, are tuning the parameters for decision and prediction. And throughout the paper, we called attention to computational models of learning in artificial agents that explain the cognitive learning in biologic agents. We ended by sharing the theory that Supervised Learning, Reinforcement Learning, and Unsupervised Learning, are likely to realize the functions of the cerebellum, the basal ganglia, and the cerebral cortex, respectively

It is our view that the RL algorithms are the most accurate representation of human learning, and that they can approximate human psychology and act as a predictor for their actions in a Markovian environment. The remarkable advances and the large body of work and literature on RL have paved the way to computational neuroscientists to bridge RL and behavioral neuroscience, but some other facets of RL still command attention.

One such very promising study is “learning by transfer” or social inference, in which RL agents learn by imitating the behaviour of other agents they socialize with. This will not necessarily lead to an optimal policy solution, but it definitely bears resemblance to

the psychological strategies of humans in societies: finding representations that are heuristic for a given problem rather than seeking a perfect solution. From the way this latter sentence is termed, we can already sense a similarity with OCD behaviour. Patients with OCD are known for favoring perfection over execution. Their representation of the world is driven solely from internal states, rather that from an observation of cues and properties of other individuals and the mimicry of their behavior [Seow, T.X.F., Gillan, C.M. (2020) https://www.nature.com/articles/s41598-020-59646-4]. It is therefore very plausible that an agent is displaying OCD symptoms, if this agent takes no observation from other agents surrounding it and rely solely on deriving an optimal solution (e.g. taking only paths with lowest cost in crowd navigation of mazes). OCD is also characterized by an excessive habit formation, which in RL would translate to a recalcitrant policy and thus the agent inability to adapt to new environments (i.e. no Meta-Learning). This maladapted behavior is reminiscent of the egocentric navigational strategy, where the subject when met with an altered topology of the environment during the probe phase, will prefer taking the exact route that registered to his or her point of view during training, rather that adapting to this spatial alteration. To illustrate that, our group of researcher intend to setup experiments that will diagnose OCD in robot agents based on these hypothesis. If validated in robot agents, this experimental platform will enable computational psychiatrists to tune the RL and LSTM model parameters until we observe a fading of the OCD behavior or a resurgence of it thereof. This direction of study is inspired from Erich Fromm's *Social Character Theory* which was introduced in the mid 90's, and which assumes that "at the group scale, psychological traits are no longer defined by the complete image of individual psychic, rather it is based on the common psychological features across the group members".

~~The introduction should briefly place the study in a broad context and highlight why it is important. It should define the purpose of the work and its significance. The current state of the research field should be carefully reviewed and key publications cited. Please highlight controversial and diverging hypotheses when necessary. Finally, briefly mention the main aim of the work and highlight the principal conclusions. As far as possible, please keep the introduction comprehensible to scientists outside your particular field of research. References should be numbered in order of appearance and indicated by a numeral or numerals in square brackets—e.g., [1] or [2,3], or [4–6]. See the end of the document for further details on references.~~

~~2. Materials and Methods~~

~~The Materials and Methods should be described with sufficient details to allow others to replicate and build on the published results. Please note that the publication of your manuscript implicates that you must make all materials, data, computer code, and protocols associated with the publication available to readers. Please disclose at the submission stage any restrictions on the availability of materials or information. New methods and protocols should be described in detail while well-established methods can be briefly described and appropriately cited.~~

~~Research manuscripts reporting large datasets that are deposited in a publicly available database should specify where the data have been deposited and provide the relevant accession numbers. If the accession numbers have not yet been obtained at the time of submission, please state that they will be provided during review. They must be provided prior to publication.~~

~~Interventionary studies involving animals or humans, and other studies that require ethical approval, must list the authority that provided approval and the corresponding ethical approval code.~~

~~3. Results~~

~~This section may be divided by subheadings. It should provide a concise and precise description of the experimental results, their interpretation, as well as the experimental conclusions that can be drawn.~~

~~*3.1. Subsection*~~

~~3.1.1. Subsubsection~~

~~Bulleted lists look like this:~~

- ~~First bullet;~~
- ~~Second bullet;~~
- ~~Third bullet.~~

~~Numbered lists can be added as follows:~~

1. ~~First item;~~
2. ~~Second item;~~
3. ~~Third item.~~

~~The text continues here.~~

~~*3.2. Figures, Tables and Schemes*~~

~~All figures and tables should be cited in the main text as Figure 1, Table 1, etc.~~

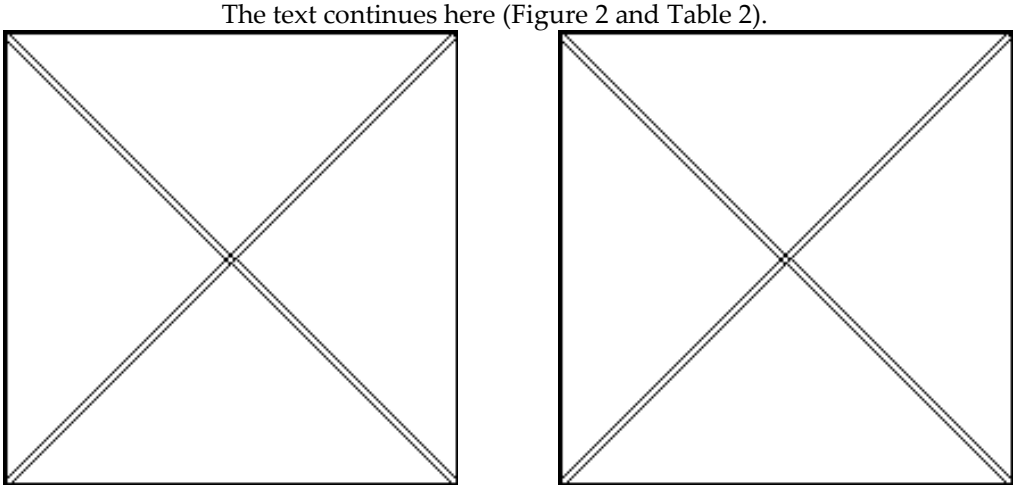
Figure 1. This is a figure. Schemes follow the same formatting.

Table 1. This is a table. Tables should be placed in the main text near to the first time they are cited.

Title 1	Title 2	Title 3
entry 1	data	data
entry 2	data	data [1]

[1] Tables may have a footer.

The text continues here (Figure 2 and Table 2).



(**a**) (**b**)

Figure 2. This is a figure. Schemes follow another format. If there are multiple panels, they should be listed as: (**a**) Description of what is contained in the first panel; (**b**) Description of what is contained in the second panel. Figures should be placed in the main text near to the first time they are cited. A caption on a single line should be centered.

Table 2. This is a table. Tables should be placed in the main text near to the first time they are cited.

Title 1	Title 2	Title 3	Title 4
	data	data	data
	data	data	data
	data	data	data
	data	data	data
	data	data	data
	data	data	data
	data	data	data
	data	data	data
	data	data	data
	data	data	data
	data	data	data
	data	data	data

~~*3.3. Formatting of Mathematical Components*~~

~~This is example 1 of an equation:~~

$$~~a = 1,~~ \quad (1)$$

~~the text following an equation need not be a new paragraph. Please punctuate equations as regular text.~~

~~This is example 2 of an equation:~~

$$~~a = b + c + d + e + f + g + h + i + j + k + l + m + n + o + p + q + r + s + t + u + v + w + x + y + z~~ \quad (2)$$

~~the text following an equation need not be a new paragraph. Please punctuate equations as regular text.~~

~~Theorem-type environments (including propositions, lemmas, corollaries etc.) can be formatted as follows:~~

~~**Theorem 1.** *Example text of a theorem. Theorems, propositions, lemmas,* etc. *should be numbered sequentially (i.e., Proposition 2 follows Theorem 1). Examples or Remarks use the same formatting, but should be numbered separately, so a document may contain Theorem 1, Remark 1 and Example 1.*~~

~~The text continues here. Proofs must be formatted as follows:~~

~~**Proof of Theorem 1.** Text of the proof. Note that the phrase "of Theorem 1" is optional if it is clear which theorem is being referred to. Always finish a proof with the following symbol. □~~

~~The text continues here.~~

~~4. Discussion~~

~~Authors should discuss the results and how they can be interpreted from the perspective of previous studies and of the working hypotheses. The findings and their implications should be discussed in the broadest context possible. Future research directions may also be highlighted.~~

~~5. Conclusions~~

~~This section is not mandatory but can be added to the manuscript if the discussion is unusually long or complex.~~

~~**6. Patents**~~

~~This section is not mandatory but may be added if there are patents resulting from the work reported in this manuscript.~~

Supplementary Materials: The following are available online at www.mdpi.com/xxx/s1, Figure S1: title, Table S1: title, Video S1: title.

Author Contributions: For research articles with several authors, a short paragraph specifying their individual contributions must be provided. The following statements should be used "Conceptualization, X.X. and Y.Y.; methodology, X.X.; software, X.X.; validation, X.X., Y.Y. and Z.Z.; formal analysis, X.X.; investigation, X.X.; resources, X.X.; data curation, X.X.; writing—original draft preparation, X.X.; writing—review and editing, X.X.; visualization, X.X.; supervision, X.X.; project administration, X.X.; funding acquisition, Y.Y. All authors have read and agreed to the published version of the manuscript." Please turn to the CRediT taxonomy for the term explanation. Authorship must be limited to those who have contributed substantially to the work reported.

Funding: Please add: "This research received no external funding" or "This research was funded by NAME OF FUNDER, grant number XXX" and "The APC was funded by XXX". Check carefully that the details given are accurate and use the standard spelling of funding agency names at https://search.crossref.org/funding. Any errors may affect your future funding.

Data Availability Statement: In this section, please provide details regarding where data supporting reported results can be found, including links to publicly archived datasets analyzed or generated during the study. Please refer to suggested Data Availability Statements in section "MDPI Research Data Policies" at https://www.mdpi.com/ethics. You might choose to exclude this statement if the study did not report any data.

Acknowledgments: In this section, you can acknowledge any support given which is not covered by the author contribution or funding sections. This may include administrative and technical support, or donations in kind (e.g., materials used for experiments).

Conflicts of Interest: Declare conflicts of interest or state "The authors declare no conflict of interest." Authors must identify and declare any personal circumstances or interest that may be perceived as inappropriately influencing the representation or interpretation of reported research results. Any role of the funders in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript, or in the decision to publish the results must be declared in this section. If there is no role, please state "The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results".

Appendix A

The appendix is an optional section that can contain details and data supplemental to the main text—for example, explanations of experimental details that would disrupt the flow of the main text but nonetheless remain crucial to understanding and reproducing the research shown; figures of replicates for experiments of which representative data is shown in the main text can be added here if brief, or as Supplementary data. Mathematical proofs of results not central to the paper can be added as an appendix.

Appendix B

All appendix sections must be cited in the main text. In the appendices, Figures, Tables, etc. should be labeled starting with "A"—e.g., Figure A1, Figure A2, etc.

References

References must be numbered in order of appearance in the text (including citations in tables and legends) and listed individually at the end of the manuscript. We recommend preparing the references with a bibliography software package, such as EndNote, ReferenceManager or Zotero to avoid typing mistakes and duplicated references. Include the digital object identifier (DOI) for all references where available.

Citations and references in the Supplementary Materials are permitted provided that they also appear in the reference list here.

In the text, reference numbers should be placed in square brackets [ ] and placed before the punctuation; for example [1], [1–3] or [1,3]. For embedded citations in the text with pagination, use both parentheses and brackets to indicate the reference number and page numbers; for example [5] (p. 10), or [6] (pp. 101–105).

1. Author 1, A.B.; Author 2, C.D. Title of the article. *Abbreviated Journal Name* **Year**, *Volume*, page range.
2. Author 1, A.; Author 2, B. Title of the chapter. In *Book Title*, 2nd ed.; Editor 1, A., Editor 2, B., Eds.; Publisher: Publisher Location, Country, 2007; Volume 3, pp. 154–196.
3. Author 1, A.; Author 2, B. *Book Title*, 3rd ed.; Publisher: Publisher Location, Country, 2008; pp. 154–196.
4. Author 1, A.B.; Author 2, C. Title of Unpublished Work. *Abbreviated Journal Name* stage of publication (under review; accepted; in press).
5. Author 1, A.B. (University, City, State, Country); Author 2, C. (Institute, City, State, Country). Personal communication, 2012.
6. Author 1, A.B.; Author 2, C.D.; Author 3, E.F. Title of Presentation. In Title of the Collected Work (if available), Proceedings of the Name of the Conference, Location of Conference, Country, Date of Conference; Editor 1, Editor 2, Eds. (if available); Publisher: City, Country, Year (if available); Abstract Number (optional), Pagination (optional).
7. Author 1, A.B. Title of Thesis. Level of Thesis, Degree-Granting University, Location of University, Date of Completion.
8. Title of Site. Available online: URL (accessed on Day Month Year).

9. Bibliography

1. Dezfouli A, Griffiths K, Ramos F, Dayan P, Balleine BW. 2019. Models that learn how humans learn: The case of decision-making and its disorders. *PLoS Comput Biol* **15**: e1006903.
2. Dezfouli A, Nock R, Dayan P. 2020. Adversarial vulnerabilities of human decision-making. *Proc Natl Acad Sci USA* **117**: 29221–29228.
3. Mnih V, Badia AP, Mirza M, Graves A, Lillicrap TP, Harley T, Silver D, Kavukcuoglu K. 2016. Asynchronous Methods for Deep Reinforcement Learning. *arXiv*.
4.