

# Apprentissage par renforcement avec répétition d'expériences

Paul Ecoffet

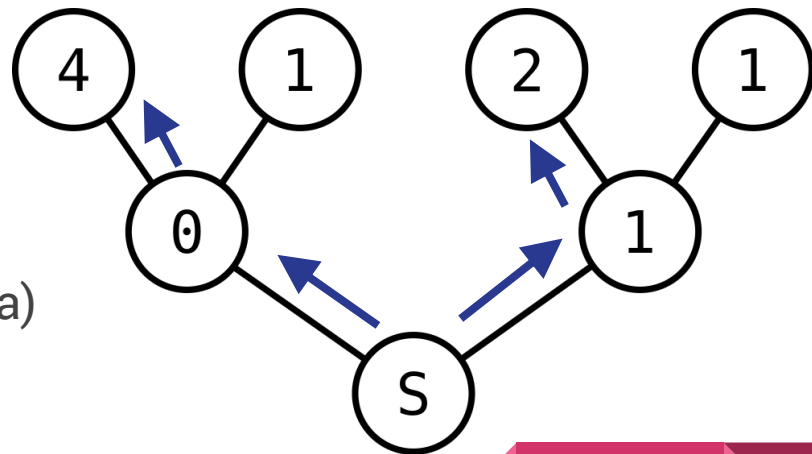
Soutenance de ministage M1 — Cogmaster  
17/06/2016

# Apprentissage par renforcement

Recherche de la meilleure suite d'opérations pour maximiser une récompense

→ Calcul d'une politique

→ Calcul d'une fonction de valeur  
d'un état :  $V(s)$   
ou d'une paire (état, action):  $Q(s, a)$



# Différence Temporelle

Association de valeur à chaque état en fonction de l'observation que l'agent vient de faire du monde.

$$\begin{aligned}\tilde{V}_1(s_0) &\leftarrow \tilde{V}_0(s_0) + \overbrace{\alpha(r_1 + \gamma\tilde{V}_0(s_1) - \tilde{V}_0(s_0))}^{\delta} \\ \tilde{V}_2(s_1) &\leftarrow \tilde{V}_1(s_1) + \alpha(r_2 + \gamma\tilde{V}_1(s_2) - \tilde{V}_1(s_1)) \\ \tilde{V}_3(s_2) &\leftarrow \tilde{V}_2(s_2) + \alpha(r_3 + \gamma\tilde{V}_2(s_3) - \tilde{V}_2(s_2))\end{aligned}$$

# Répétition de séquences

Répétition de séquence d'expériences à l'aide de connaissances actuelles pour améliorer l'estimation de la valeur des états.

$$\tilde{V}'_1(s_0) \leftarrow \tilde{V}_0(s_0) + \alpha(r_1 + \gamma \tilde{V}_*(s_1) - \tilde{V}_0(s_0))$$

$$\tilde{V}'_2(s_1) \leftarrow \tilde{V}'_1(s_1) + \alpha(r_2 + \gamma \tilde{V}_*(s_2) - \tilde{V}'_1(s_1))$$

$$\tilde{V}'_3(s_2) \leftarrow \tilde{V}'_2(s_2) + \alpha(r_3 + \gamma \tilde{V}_*(s_3) - \tilde{V}'_2(s_2))$$

Plusieurs algorithmes d'apprentissage par renforcement utilisent du replay et des variations de ce calcul

# Le Replay en neurobiologie

Redéclenchement de séquences d'activation de cellules de lieu sur certaines périodes (durant Sharp Wave Ripples complexes) :

- Sommeil
- Immobile

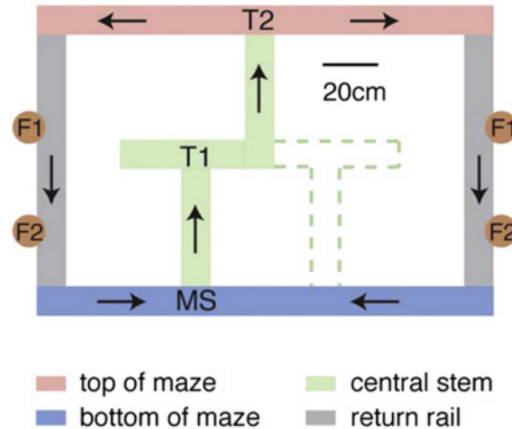
Plusieurs types de séquences rejoués, avec origine et fin de séquences non expliqués

- Forward
- Backward
- Shortcut

# Le Replay en neurobiologie

Gupta et al., 2010 étudie le replay chez le rat

→ Étude de l'activation des cellules de lieu lors que le rat mange sa récompense



# Le Replay en neurobiologie

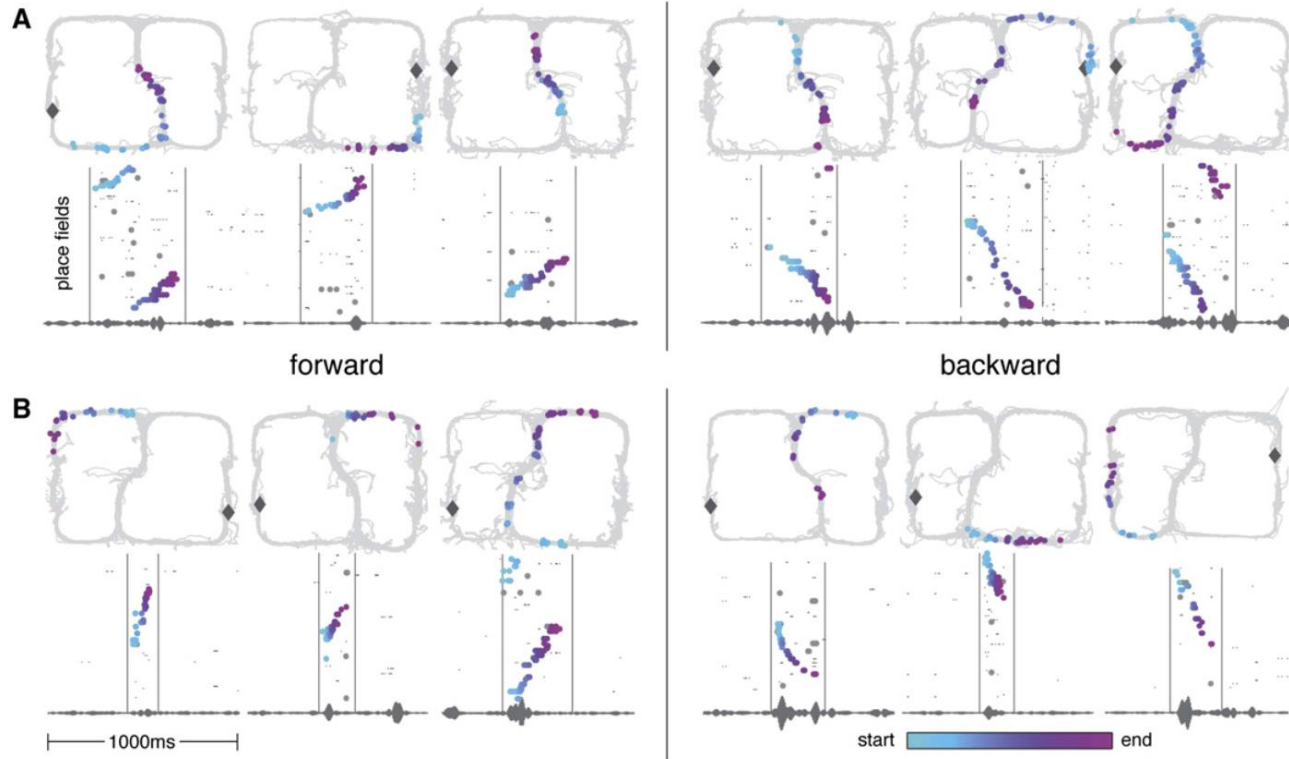


Figure issue de Gupta et al., 2010

# Le Replay en neurobiologie

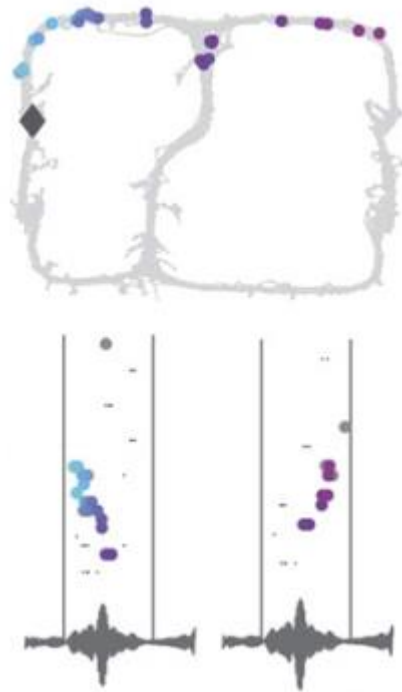


Figure issue de Gupta et al., 2010



# Étudier ces séquences aux travers d'algorithmes

- Les différents types d'algorithmes d'apprentissage par renforcement avec replays peuvent-ils reproduire les séquences de replays observées par Gupta et al. ?
- Peut-on utiliser ces algorithmes pour formuler des prédictions quant aux séquences de replays observées ?



# Linear Dyna

Idée : Choisir les bons états à générer pour accélérer la convergence

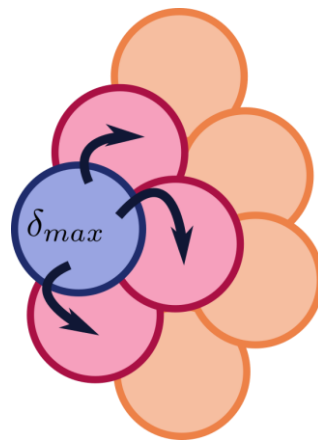
Choisir éléments déjà visités avec une erreur de prédiction forte très efficace

→ Créer une liste des états visités et on les trie par erreur de prédiction  $\delta$

# Linear Dyna

Lors du replay:

- Sélection de l'état avec la plus forte erreur et mise à jour de ses voisins à l'aide du modèle
- On ajoute à la liste chaque voisin avec pour priorité l'erreur de prédiction



# Linear Dyna

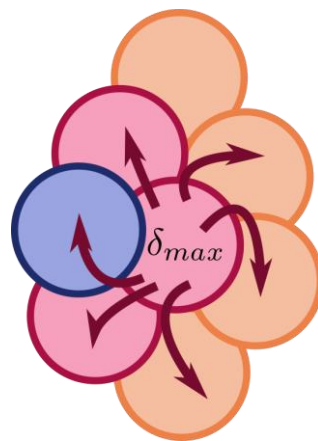
Lors du replay:

→ Sélection de l'état avec la plus forte erreur et mise à jour de ses voisins à l'aide du modèle

→ On ajoute à la liste chaque voisin avec pour priorité l'erreur de prédiction

→ On répète

$\delta$  interprété comme surprise et corrélé à dopamine dans la littérature




# Expérience

Simulation de l'expérience de Gupta à l'aide d'un Linear Dyna avec Replay (MG Prioritized Sweeping)

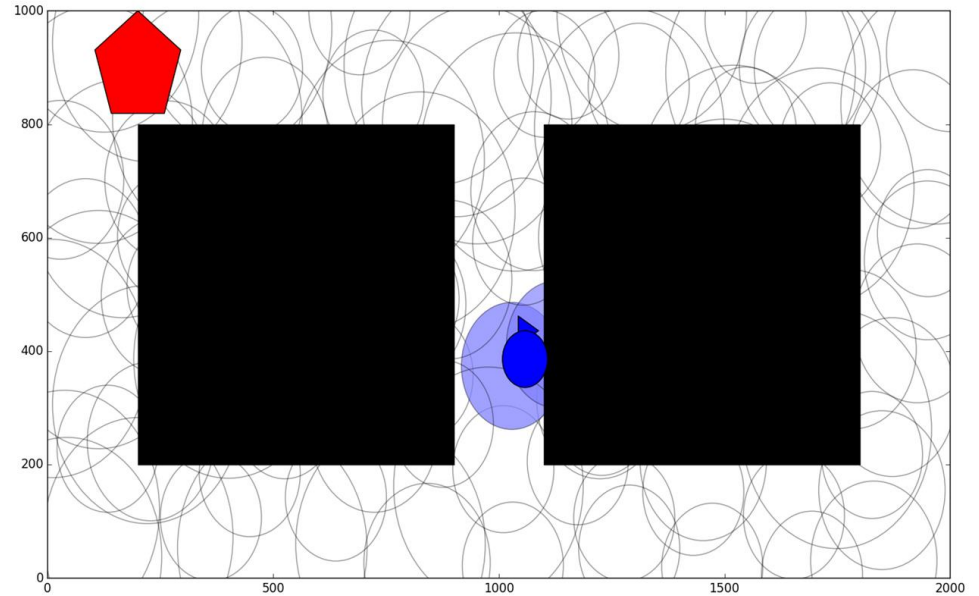
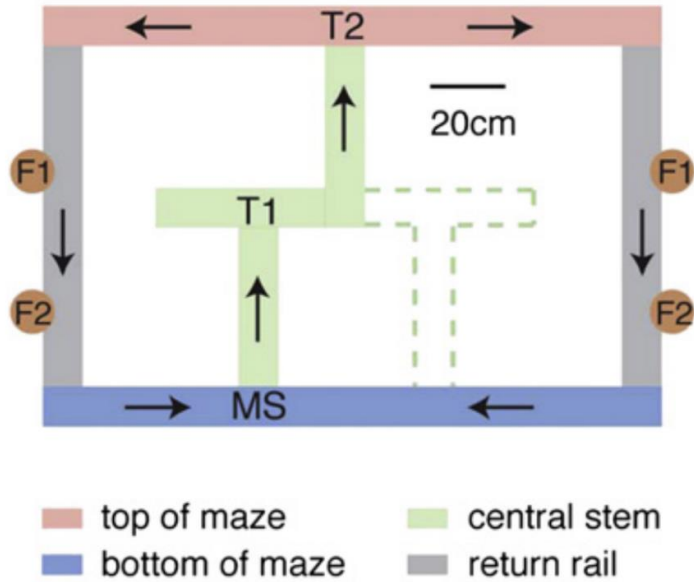
## Hypothèse :

Les replays générés par l'exécution de Linear Dyna vont former des séquences et elles seront comparables aux séquences de replay observées par Gupta et al.

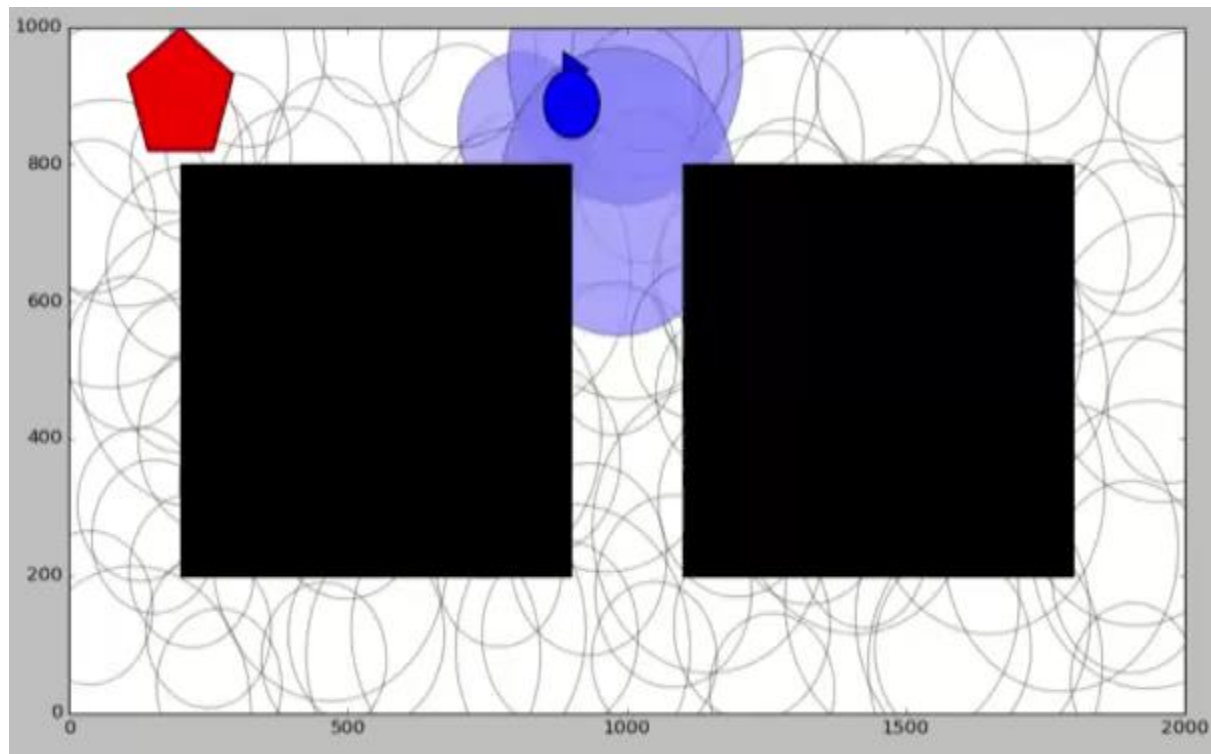
→ Proposer nouvelles prédictions pour tester robustesse du modèle



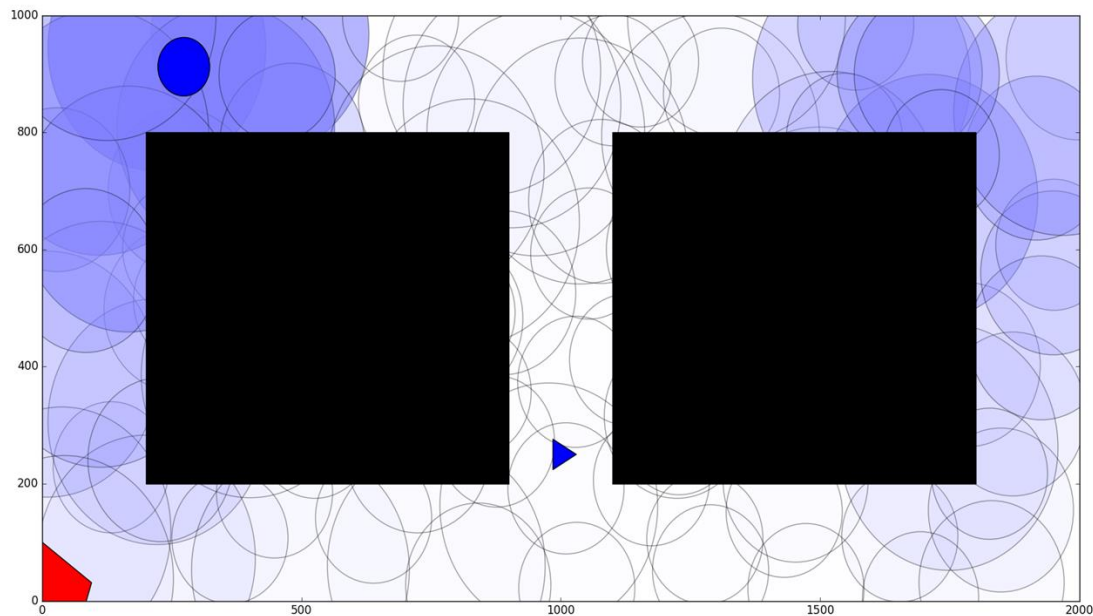
# Expérience



# Expérience



# Résultats

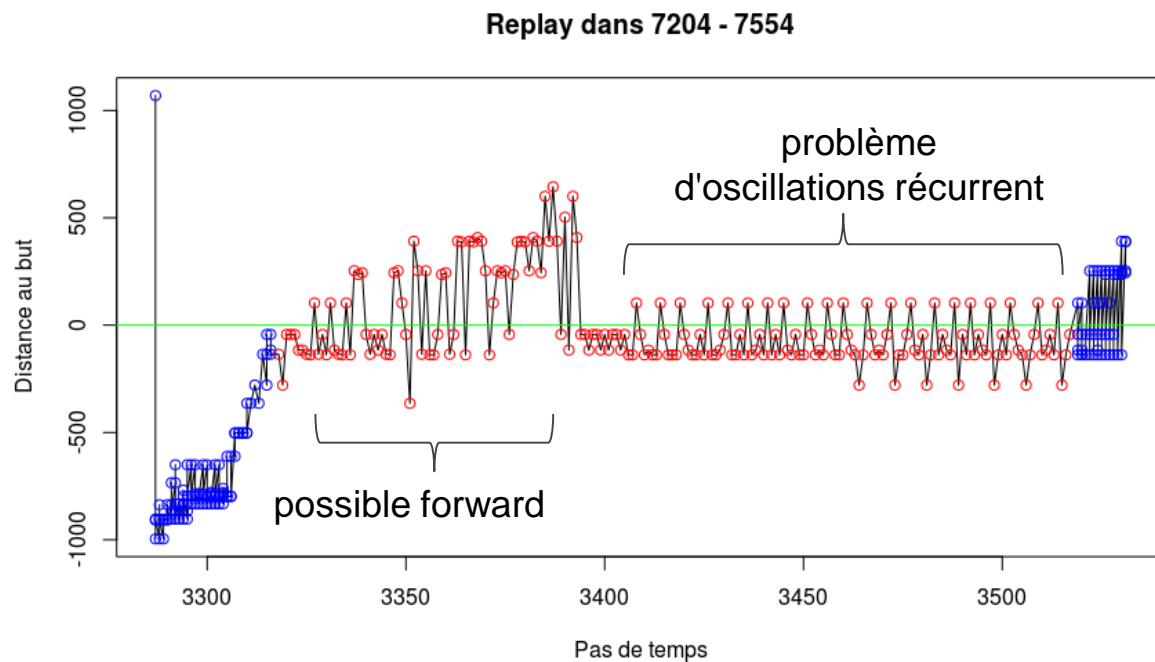


Fonction de valeur par état

bleu : valeur haute comparé à l'ensemble



# Résultats



# Discussion

Pas de résultats clairement visibles pour l'instant

→ Analyse des résultats encore peu poussée

Fortes oscillations entre les états très proches de la récompense

Replay de petite étendue

Beaucoup de paramètres libres, dont les valeurs changent le comportement ou la convergence de l'algorithme ( $\alpha$ ,  $\gamma$ , nombre de cellules de lieu, taille du champ récepteur, valeur et bruit de la récompense, nombre de replay)





Merci

# Annexes

# TD(0)

---

**Algorithm 1:** Canonical TD(0)

---

**Input:** current  $\tilde{V}$  (usually  $\vec{0}$ ), first state  $s$ , policy  $\pi$ , learning rate  $\alpha$ ,  
discount factor  $\gamma$

**Output:** Updated  $\tilde{V}$

**while**  $s$  *is not end* **do**

$a \leftarrow$  action according to  $\pi$  for  $s$

    Do  $a$ , observe the reward  $r$  and the new state  $s'$

$\delta \leftarrow r + \gamma \tilde{V}(s') - \tilde{V}(s)$

$\tilde{V}(s) \leftarrow \tilde{V}(s) + \alpha \delta$

$s \leftarrow s'$

**end**

---



# TD(0) avec répétition de séquence

---

**Algorithm 2:** TD(0) with Replay

---

**Input:**  $\tilde{V}_{init}$  (usually  $\vec{0}$ ), first state  $s$ , policy  $\pi$ , learning rate  $\alpha$ , discount factor  $\gamma$ , computational budget  $k$

**Output:** Updated  $\tilde{V}$

$\tilde{V} \leftarrow \tilde{V}_{init}$

$\mathcal{M} \leftarrow \emptyset$  // The list of observed samples

**while**  $s$  *is not end* **do**

$a \leftarrow$  action according to  $\pi$  for  $s$

    Do  $a$ , observe the reward  $r$  and the new state  $s'$

    append  $(s, r, s')$  to  $\mathcal{M}$

**for**  $i \leftarrow 1$  to  $k$  **do**

$\tilde{V}_* \leftarrow \tilde{V}$

$\tilde{V} \leftarrow \tilde{V}_{init}$

**foreach**  $(s, r, s') \in \mathcal{M}$  (from oldest to newest) **do**

$\delta \leftarrow r + \gamma \tilde{V}_*(s') - \tilde{V}(s)$

$\tilde{V}(s) \leftarrow \tilde{V}(s) + \alpha \delta$

**end**

**end**

$s \leftarrow s'$

**end**

---

# Linear Dyna with MG Prioritized Sweeping

---

**Algorithm 3** : Linear Dyna with MG prioritized sweeping  
(policy evaluation)

---

Obtain initial  $\phi, \theta, F, b$

For each time step:

Take action  $a$  according to the policy. Receive  $r, \phi'$

$$\delta \leftarrow r + \gamma \theta^\top \phi' - \theta^\top \phi$$

$$\theta \leftarrow \theta + \alpha \delta \phi$$

$$F \leftarrow F + \alpha (\phi' - F\phi) \phi^\top$$

$$b \leftarrow b + \alpha (r - b^\top \phi) \phi$$

For all  $i$  such that  $\phi(i) \neq 0$ :

Put  $i$  on the PQueue with priority  $|\delta \phi(i)|$

Repeat  $p$  times while PQueue is not empty:

$i \leftarrow \text{pop the PQueue}$

For all  $j$  such that  $F^{ij} \neq 0$ :

$$\delta \leftarrow b(j) + \gamma \theta^\top F e_j - \theta(j)$$

$$\theta(j) \leftarrow \theta(j) + \alpha \delta$$

Put  $j$  on the PQueue with priority  $|\delta|$

$$\phi \leftarrow \phi'$$

---

McMahan and Gordon (2005)

Avec vecteur : Sutton et al. (2012)

pseudo-code de Sutton et al. (2012)

# Forgetful LSTD

---

**Algorithm 7** Forgetful LSTD( $\lambda$ )

---

**INPUT:**  $\alpha, \beta, \lambda, k, \theta_{init}, \mathbf{d}_{init}, A_{init}$

*// For replay-equivalence, use:*

*//  $\beta \leftarrow \alpha, A_{init} \leftarrow \mathcal{I}/\alpha, \mathbf{d}_{init} \leftarrow \theta_{init}/\alpha$*

$\theta \leftarrow \theta_{init}, \mathbf{d} \leftarrow \mathbf{d}_{init}, A \leftarrow A_{init}$

obtain initial  $\phi$

$e \leftarrow \mathbf{0}$

Loop:

    obtain next feature vector  $\phi', \gamma$  and reward  $R$

$e \leftarrow (\mathcal{I} - \beta \phi \phi^\top) e + \phi$

$A \leftarrow (\mathcal{I} - \beta \phi \phi^\top) A + e(\phi - \gamma \phi')^\top$

$\mathbf{d} \leftarrow (\mathcal{I} - \beta \phi \phi^\top) \mathbf{d} + eR$

$e \leftarrow \gamma \lambda e$

    Repeat  $k$  times:

$\theta \leftarrow \theta + \alpha(\mathbf{d} - A\theta)$

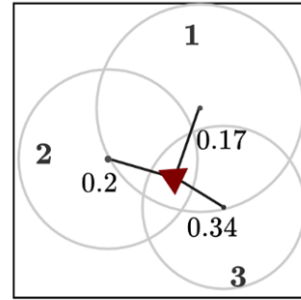
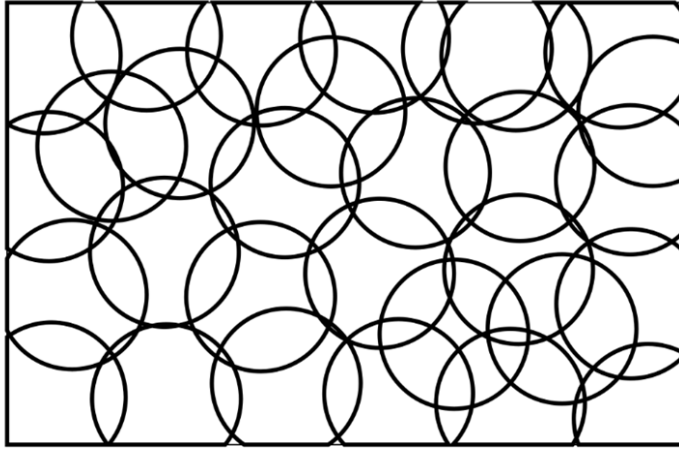
$\phi \leftarrow \phi'$

---

issu de Van Seijen & Sutton (2015)



# Fonctionnement conversion en vecteur de caractéristiques



$$\phi = \begin{pmatrix} 0.17 \\ 0.2 \\ 0.34 \end{pmatrix}$$

# Apprentissage par renforcement

Mettre à jour la valeur d'un état

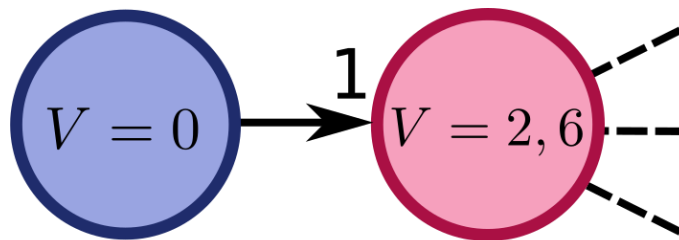
$$V(s_t) = r + \gamma V(s_{t+1})$$

$r$ : Récompense obtenu dans la transition  $s_t \rightarrow s_{t+1}$

$\gamma$ : Facteur de dévaluation

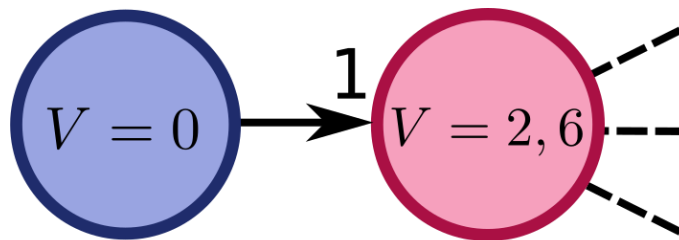
# Différence Temporelle

Association de valeur à chaque état en fonction de l'observation que l'agent vient de faire du monde.



# Différence Temporelle

Association de valeur à chaque état en fonction de l'observation que l'agent vient de faire du monde.



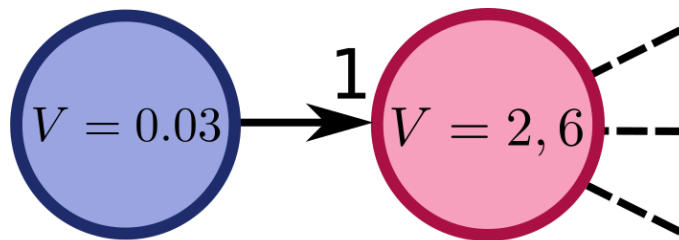
$$\delta = r + \gamma V(\text{rouge}) - V(\text{bleu})$$

$$\delta = 1 + 0,9 \cdot 2,6 - 0 = 3,34$$

$\delta$  : Erreur de prédiction

# Différence Temporelle

Association de valeur à chaque état en fonction de l'observation que l'agent vient de faire du monde.



$$\delta = r + \gamma V(\text{rouge}) - V(\text{bleu})$$

$$V(\text{bleu}) = V(\text{bleu}) + \alpha \delta$$

$\alpha$  : Taux d'apprentissage

# Les occurrences de replay

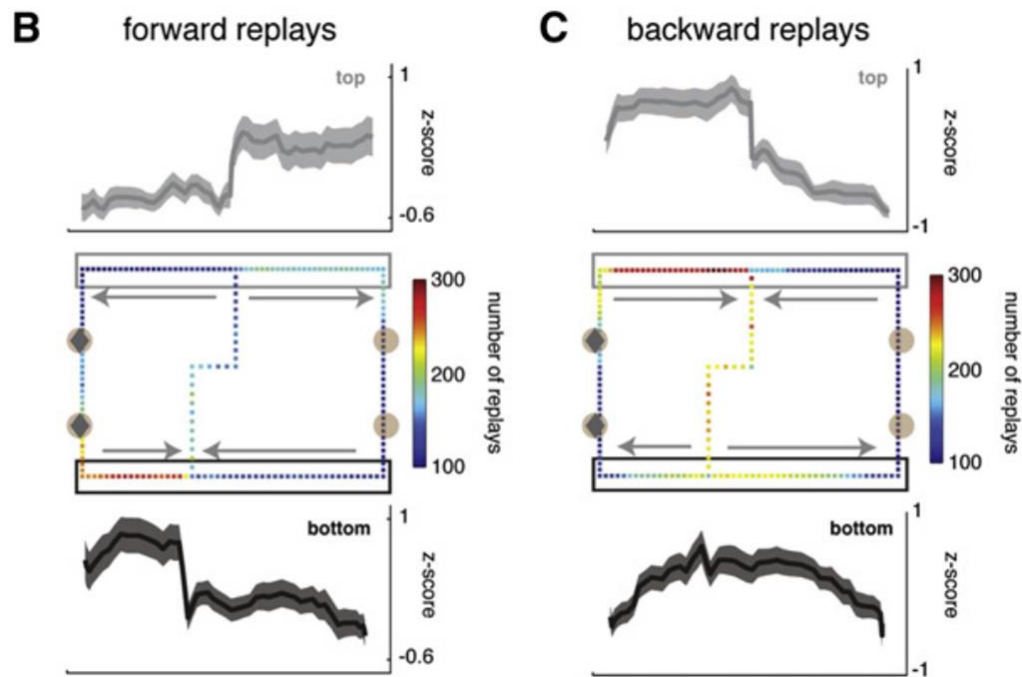


Figure issue de Gupta et al., 2010