Reinforcement learning

- Esquema motivado biológicamente
- No se separan de manera estricta los procesos de aprendizaje y ejecución

  (está todo mezclado)
- Se puede implementar con redes neuronales artificiales pero también con otros esquemas

Reinforcement learning

- El entorno es una fuente de estímulos (o receptor de acciones)
- El aprendizaje debe involucrar algún feedback de si el resultado del comportamiento es correcto o no
- Reinforcement learning (RL) es el problema que tiene un agente que debe aprender por prueba y error las interacciones con su ambiente.

No hay una regla explícita de qué es bueno y qué es malo, sino que hay que ir aprendiendo por prueba y error los comportamientos correctos

Reinforcement learning

- Hay un agente que existe en entorno descrito por in conjunto S de posibles estados, un conjunto A de posibles acciones y una **recompensa** r_t que el agente recibe a tiempo t después que tome una acción en un estado
- Alternativamente, la recompensa podría ocurrir solo despues que una **secuencia** de acciones ha sido ejecutada
- Típicamente se supone que el entorno es no-determinista
- La evaluación del agente (en término de recompensa) puede ser intercalada con el aprendizaje

Reinforcement learning

Jerga:

- El objetivo de un agente RL es maximizar la recompensa acumulada durante su tiempo de vida

- **Time step**: el agente esta en un estado, s_t, toma accion a, y se mueve al estado s_{t+1}. Después de llegar a s_{t+1} , el agente recibe la recompensa, r_t.
- ***Trial:*** esto denota un ***episodio.*** Un trial consiste de un conjunto de pasos que termina cuando el agente llega a un *estado terminal* o a un límite de tiempo (número sde pasos) predeterminado
- ***Estado Terminal (or absorbing)* –** un estado que el agente no deja, que puede incluir una recompensa o castigo final. Un ***estado objetivo*** es un ejemplo de estado terminal.

Reinforcement learning

- El sistema está definido por dos funciones
  - $\delta(s_t,a_t)=s_{t+1}$: función transición
  - $r(s_t,a_t)$: función recompensa. A veces se escribe $r(s_t,s_{t+1})$
  - Si el entorno es estocástico podemos tener $p(s_t,a_t,s_{t+1})$: probabilidad de ir al estado s_{t+1}
  - El modelo NO es conocido por el agente

    El agente solo ve ejemplos
  - Un factor de descuento es útil (recompensas futuras son menos valiosas)

    Es mejor obtener una recompensa ahora que en un tiempo

Reinforcement learning

- La política (*policy*): es un mapeo completo de cada estado en cada acción a ser tomada en ese estado
- El objetivo de RL es encontrar la *política óptima*
- La política óptima es la que maximiza la *recompensa acumulada descontada.*

$$V^{\Pi}(s_t) \equiv r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots = \sum_{i=0}^{\infty} \gamma^i r_{t+i}$$

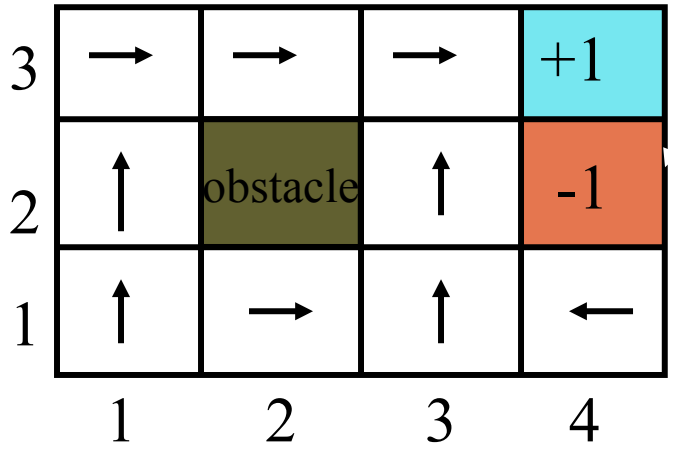
γ factor de descuento

Reinforcement learning

Ejemplo de política óptima

Estados terminales

El conjunto de flechas marcadas forma una política óptima porque partiendo de cualquier estado (cualquier cuadrado) se puede llegar a la recompensa ("+1") en el menor tiempo posible (menor nro de pasos)



Recompensa es -0.04 en todos los estados no-terminales.
Mostramos recompensa en estados terminales (4,3) and (4,2) .
Sin descuento.

Nota: ***Puede haber mas de una política óptima.***

Reinforcement learning

- La elección de las recompensas que se le da al agente puede determinar qué tan rápido aprenderá
- Por ejemplo, si se otorga una recompensa de 0,99 por cada estado que conduce directamente a la meta y una recompensa de 0 por cada otro estado, entonces se le está dando una gran cantidad de conocimientos previos a su agente y se puede aprender muy rápido. porque se requiere poco aprendizaje. En esencia, se le está enseñando al agente cómo llegar a la meta seleccionando cuidadosamente sus recompensas.
- Si se otorga recompensas relativamente iguales (por ejemplo, cerca de 0) de todos los estados excepto los estados terminales, el agente tardará mucho en aprender.

Reinforcement learning

La mayoría de los algoritmos usados actualmente están basados en:

- *Temporal difference algorithm* (Sutton, 1984)

- El agente debe estimar la *utilidad* de un estado. Esta **es la suma de las recompensas descontadas que comienzan el camino desde este estado.**

  ((utilidad: suma de todos los caminos que comienzan de un dado estado inicial))

Se transforma un problema global a un problema local

- Desde cada condición inicial se mueve al estado vecino con la máxima utilidad

Como estimar la utilidad de un estado?

Reinforcement learning

- Desde cada condición inicial se mueve al estado vecino con la máxima utilidad

  ¿Cómo determino la "utilidad" de cada estado? Parto de utilidades aleatorias pequeñas, me voy moviendo y voy actualizando la utilidad en base a la siguiente regla:

- $U(s) + \alpha(r(s,s') + \gamma U(s') - U(s)) \rightarrow U(s)$

  - $r(s,s')$: recompensa of $s \rightarrow s'$
  - $U(s')$: utilidad del estado sucesor
  - α: learning rate
  - γ: factor de descuento

  - A veces se define $\delta = r(s,s') + \gamma U(s') - U(s)$ como el *error de predicción*

    Dice cuál es la recompensa real que recibí comparado con lo que esperaba recibir

  - La utilidad es la integral temporal del error de predicci*ón*

Reinforcement learning

- Inicializar U(s) = 0 para todos los estados no terminales s. Para estados terminales, U(s) = r(s)

- Comience en un estado inicial designado s0 (Suponemos que todos los demás estados son accesibles desde s0)

- Para cada transición δ (s, a) = s’ y recompensa r(s, s’) por pasar del estado s al estado s’, hacer:

$$U(s) + \alpha (r(s, s') + \gamma U(s') - U(s)) \rightarrow U(s)$$

- Repetir el paso anterior hasta que la diferencia en los valores sucesivos (antes / después de la actualización) de U sea menor o igual a algún pequeño ε deseado (llamado convergencia).

Reinforcement learning

- Una ve que el agente ha aprendido una utilidad estimada para cada estado, puede usar esta utilidad para decidir qué acción tomar a continuación; *elegirá la acción que conduce al siguiente estado con la mayor utilidad.*
- Pero también se puede elegir un estado sub-óptimo con cierta probabilidad . Esto permite recorrer una región más amplia del espacio de políticas
- Dilema *explotación-exploración*

Reinforcement learning

RL necesita una señal que satisfaga los siguientes criterios:

- Responde a contingencias
- Afecta el aprendizaje de predicciones y acciones.
- Es esencialmente escalar
- Emite su información de forma multimodal

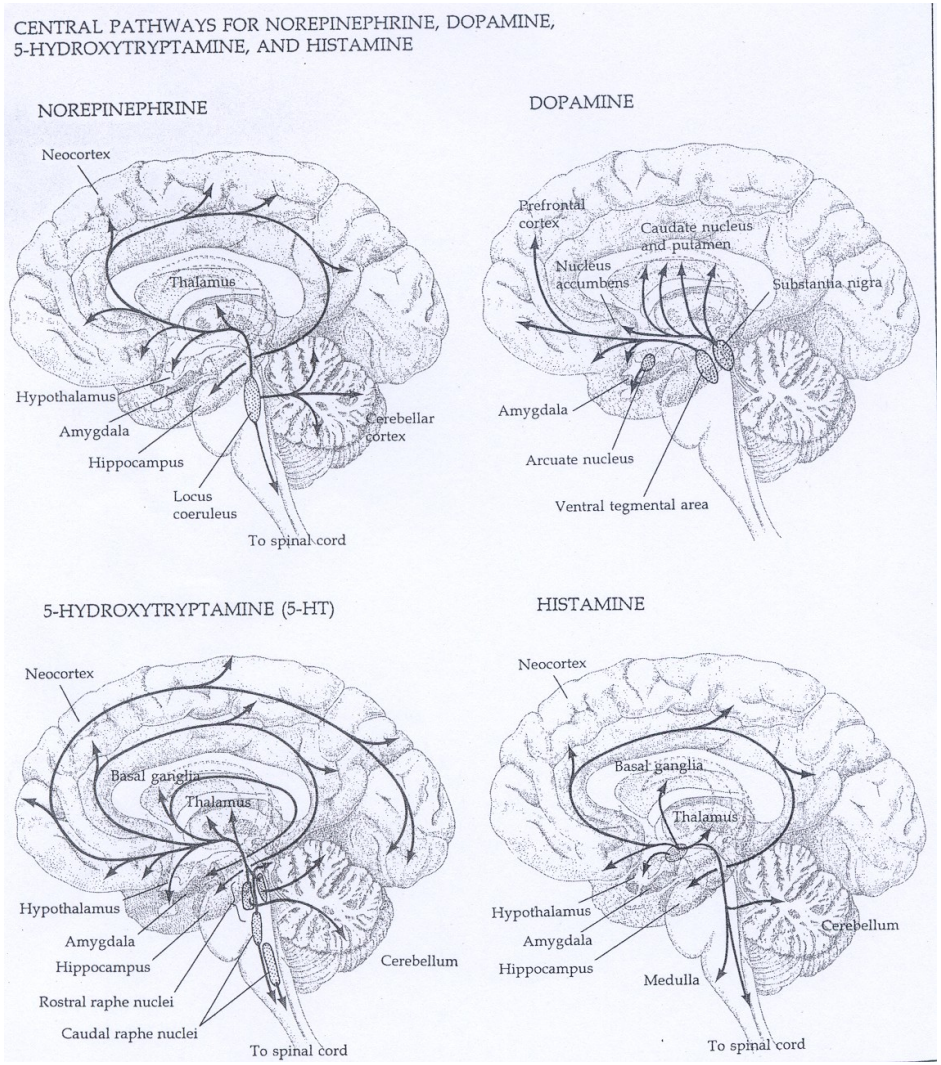
sustancias liberadas por una neurona que al ser liberados afectan a una gran porción del sistema ((de neuronas))

Los neuromoduladores verifican la mayoría de estos puntos:

- Responden a los reforzadores y a sorpresa
- Se sabe que afectan la plasticidad sináptica.
- Provienen de pequeños núcleos del cerebro medio
- Tienen una amplia arborización en todo el cerebro.

(afecta una gran región del cerebro)

Reinforcement learning



No solo ocurre en seres humanos, sino en ((cualquier)) cerebro de mamífero

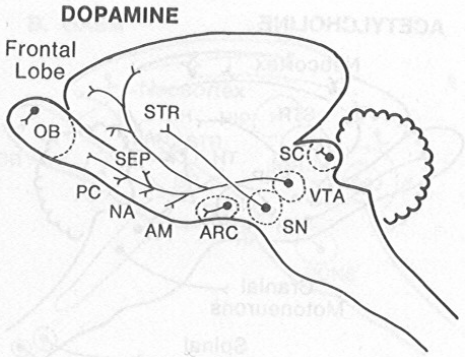


Fig. 24.10 Distribution of dopamine-containing neurons in the rat brain. For abbreviations, see legend to Fig. 24.9.

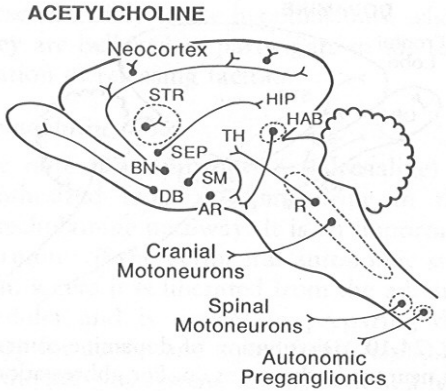
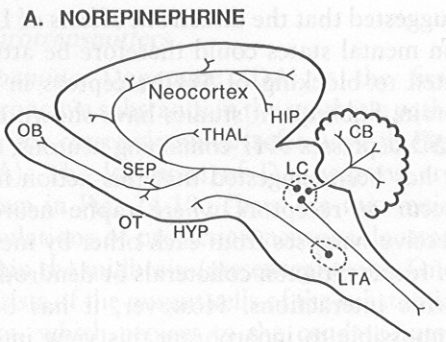


Fig. 24.11 Distribution of cholinergic cell groups and their projections in the rat brain. For abbreviations, see legend to Fig. 24.9.



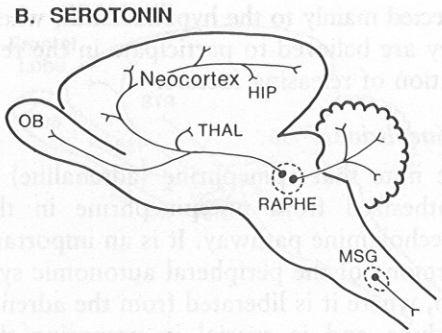
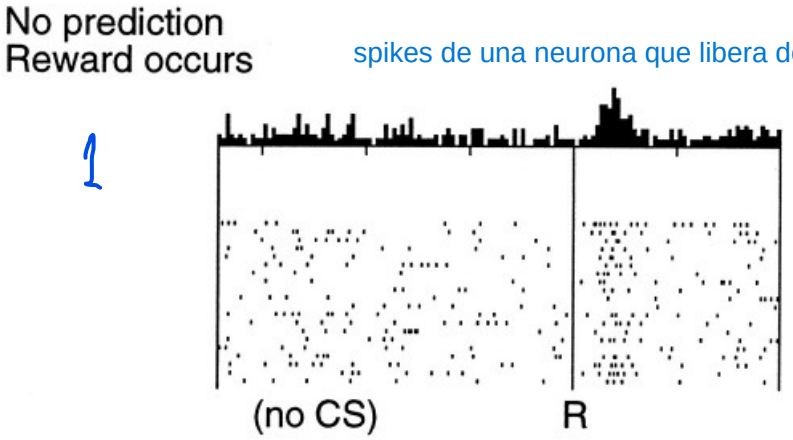


Fig. 24.9 Maps of the distribution of cell groups containing different neurotransmitters in the mammalian brain. A sagittal view of the rat brain is shown for this and succeeding figures. **A.** Distribution of norepinephrine-containing neurons and their axonal projections. **B.** Distribution of serotonin-containing neurons and their projections. These are discussed in the text under the category of central state circuits. Abbreviations for this and the following maps: AM, amygdala; AR, arcuate nucleus; ARC, arcuate nucleus; BN, basal nucleus; DB, diagonal band; DCN, deep cerebellar nuclei; DH, dorsal horn, DRG, dorsal root ganglion; EPN, endopeduncular nucleus; GP, globus pallidus; HAB, habenula; HIP, hippocampus; HYP, hypothalamus; LC, locus ceruleus; LTA, lateral tegmental area; MED, medulla; MSG, medullary serotonin group; NA, nucleus accumbens; OB, olfactory bulb; OT, olfactory tubercle; PC, pyriform cortex; PERI-V., periventricular gray; R, reticular nucleus; SC, superior colliculus; SEP, septum; SM, stria medullaris; SN, substantia nigra; STR, striatum; TH or THAL, thalamus; VTA, ventral tegmental area.

Reinforcement learning

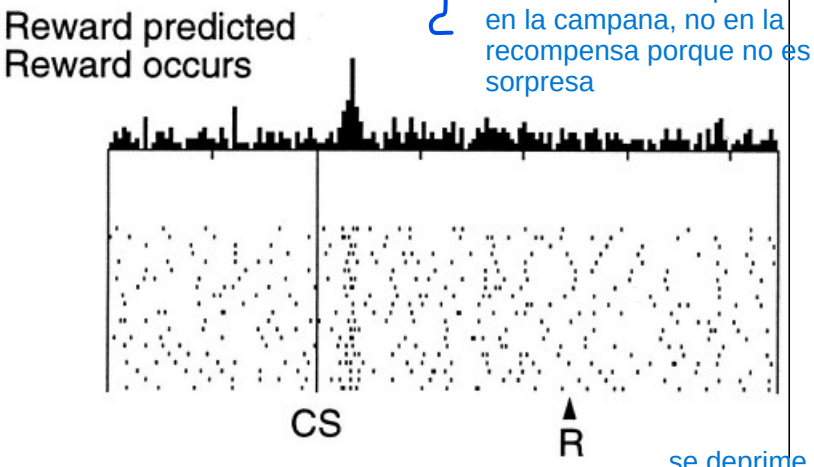
Experimento de aprendizaje condicional, como el perro de Pavlov!

- Dopamina genera una señal de recompensa *predictiva* (Schultz, 1998)



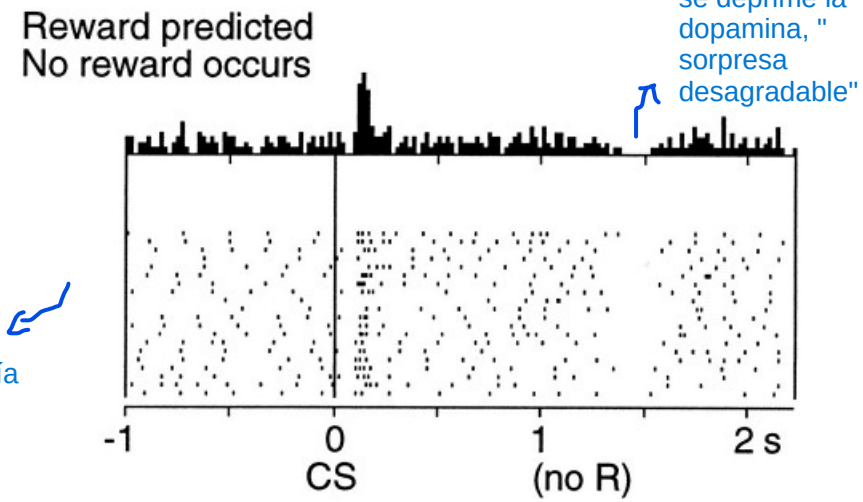
spikes de una neurona que libera dopamina

1



2

la liberación de dopamina ocurre en la campana, no en la recompensa porque no es una sorpresa



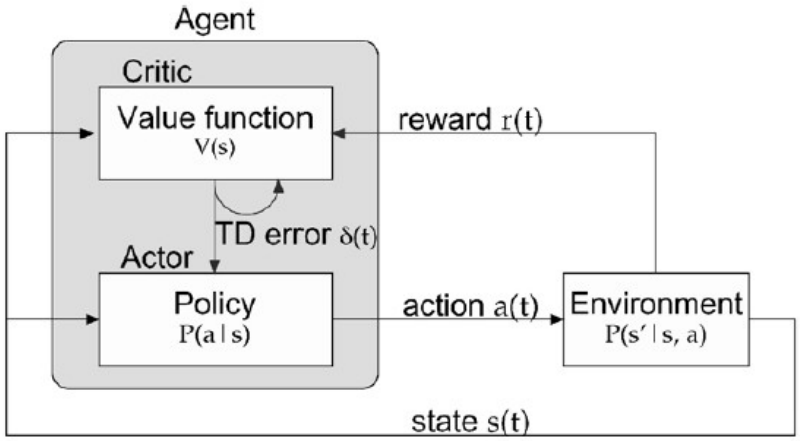
se deprime la dopamina, " sorpresa desagradable"

"Estímulo no condicionado" (no CS): no pasa nada
"Estímulo condicionado" (CS): campana
"Recompensa" (R): alimento

No solo aprendió a relacionar campana con alimento, sino que aprende el tiempo: cuánto tiempo después de la campana debería recibir el alimento. El animal es capaz de medir tiempo

Reinforcement learning

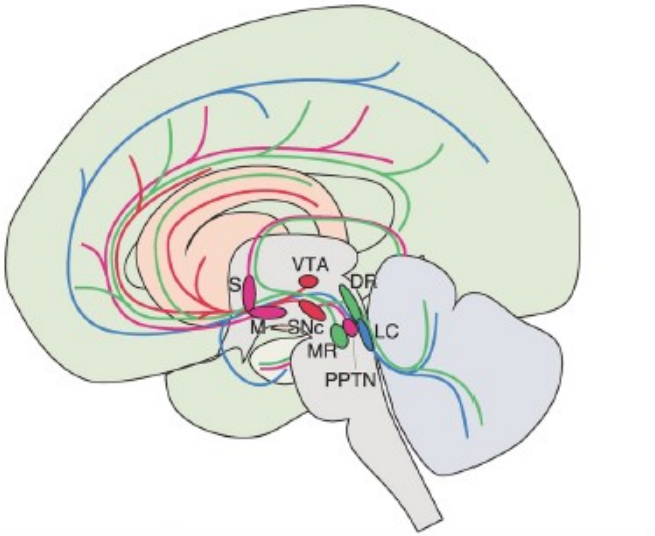
- Para que son los otros neuromoduladores? (Doja, *Metalearning and neuromodulation,* 2002) Teoría de "meta-aprendizaje". Los demás neuromoduladores tienen otro papel en el algoritmo de aprendizaje



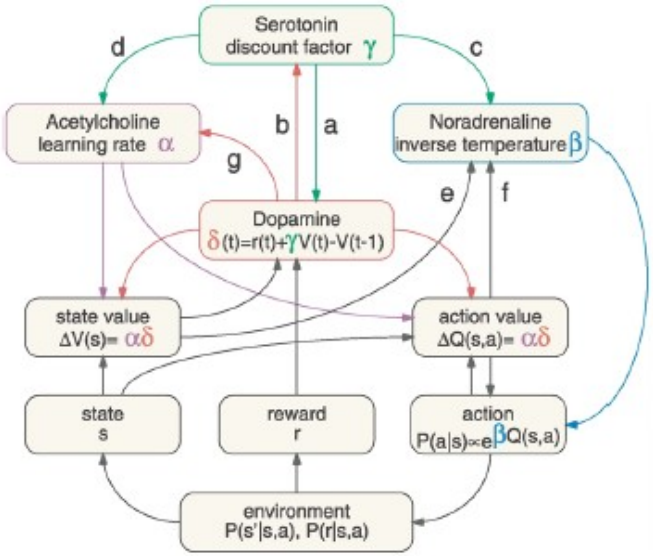
1. Dopamine signals the TD error δ.
2. Serotonin controls the discount factor γ.
3. Noradrenaline controls the inverse temperature β.
4. Acetylcholine controls the learning rate α.

Reinforcement learning

- Para que son los otros neuromoduladores? (Doja, *Metalearning and neuromodulation,* 2002)



neuromodulator	origin of projection	major target area
dopamine (DA)	substantia nigra, pars compacta (SNc) ventral tegmental area (VTA)	dorsal striatum ventral striatum frontal cortex
serotonin (5-HT)	dorsal raphe nucleus (DR) median raphe nucleus (MR)	cortex, striatum cerebellum hippocampus
noradrenaline (NA) (norepinephrine, NE)	locus coeruleus (LC)	cortex, hippocampus cerebellum
acetylcholine (ACh)	Meynert nucleus (M) medial septum (S) pedunculopontine tegmental nucleus (PPTN)	cortex, amygdala hippocampus SNc, thalamus superior colliculus



1. Dopamine represents the global learning signal for prediction of rewards and reinforcement of actions.
2. Serotonin controls the balance between short-term and long-term prediction of reward.
3. Noradrenaline controls the balance between wide exploration and focused execution.
4. Acetylcholine controls the balance between memory storage and renewal.

Reinforcement learning

Temporal difference models describe higher-order learning in humans

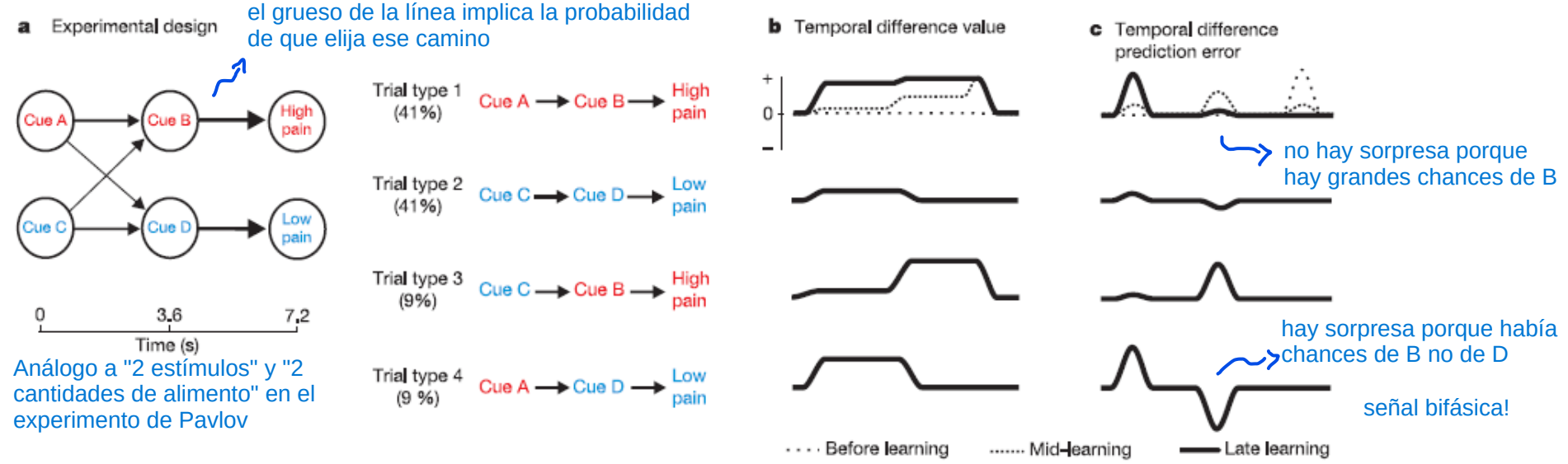
con mediciones no invasivas de actividad neuronal (usando Resonancia Magnética Funcional)

Ben Seymour[1], John P. O'Doherty[1], Peter Dayan[2], Martin Koltzenburg[3], Anthony K. Jones[4], Raymond J. Dolan[1], Karl J. Friston[1] & Richard S. Frackowiak[1,5]

NATURE | VOL 429 | 10 JUNE 2004 | www.nature.com/nature

Prediction error: $\delta = r + V(s_{t+1}) - V(s_t)$

$V(s_{t+1}) = V(s_t) + \alpha\delta$



Análogo a "2 estímulos" y "2 cantidades de alimento" en el experimento de Pavlov

Figure 1 Experimental design and temporal difference model. **a**, The experimental design expressed as a Markov chain, giving four separate trial types. **b**, Temporal difference value. As learning proceeds, earlier cues learn to make accurate value predictions (that is, weighted averages of the final expected pain). **c**, Temporal difference prediction error; during learning the prediction error is transferred to earlier cues as they acquire the ability to make predictions. In trial types 3 and 4, the substantial change in prediction elicits a large positive or negative prediction error. (For clarity, before and mid-learning are shown only for trial type 1.)

Reinforcement learning

Me fijo las regiones del cerebro según en qué caso (trial) esté

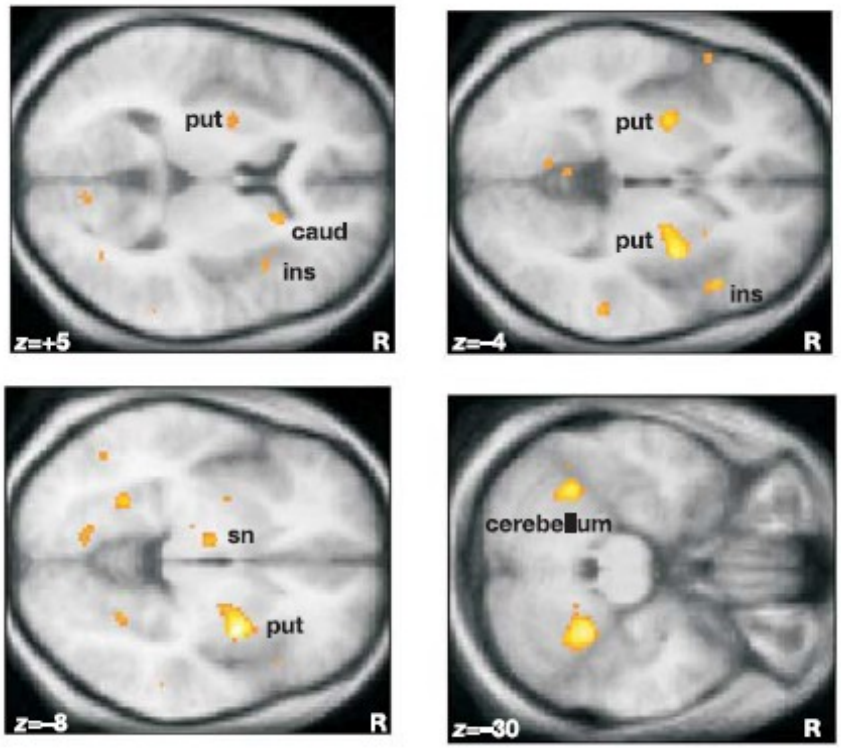
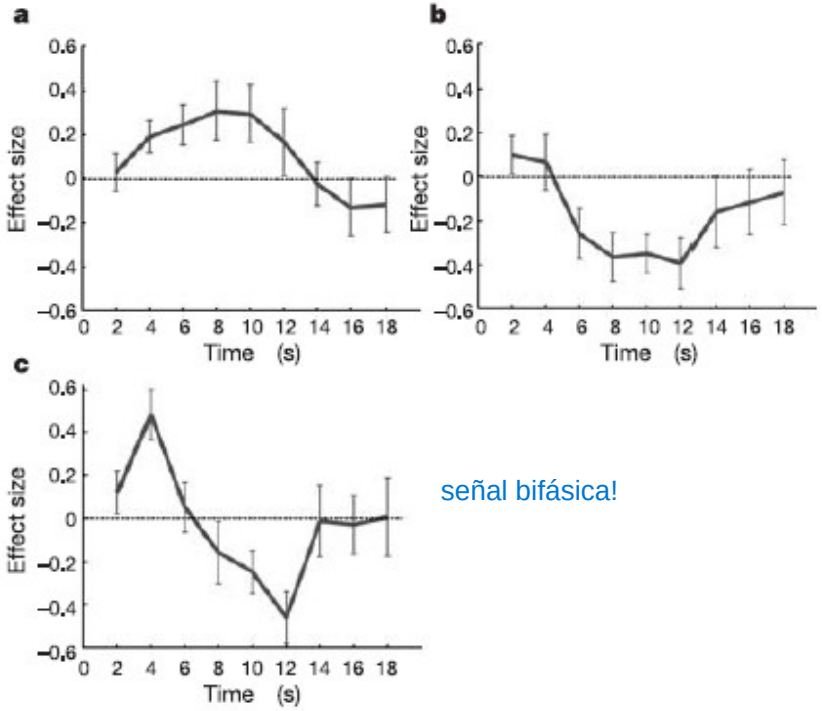


Figure 2 Temporal difference prediction error (statistical parametric maps). Areas coloured yellow/orange show significant correlation with the temporal difference prediction error. Yellow represents the greatest correlation. Peak activations (MNI coordinates and statistical *z* scores) are: right ventral putamen (put; (32, 0, −8), $z = 5.38$); left ventral putamen (put; (−30, −2, −4), $z = 3.93$); right head of caudate (caud; (18, 20, 6), $z = 3.75$); left substantia nigra (sn; (−10, −10, −8), $z = 3.52$); right anterior insula (ins; (46, 22, −4), $z = 3.71$); right cerebellum ((28, −46, −30), $z = 4.91$); and left cerebellum ((−34, −52, −28), $z = 4.42$). R indicates the right side.



señal bifásica!

Figure 3 Temporal difference prediction error (impulse responses). Time course of the impulse response (±s.e.m.) to higher-order prediction error in the right ventral putamen. **a**, Positive prediction error (contrast of trial types 3 and 2). **b**, Negative prediction error (contrast of trial types 4 and 1). **c**, Biphasic prediction error; positive at the first cue, becoming negative at the second (contrast of trial types 4 and 2).

Reinforcement learning

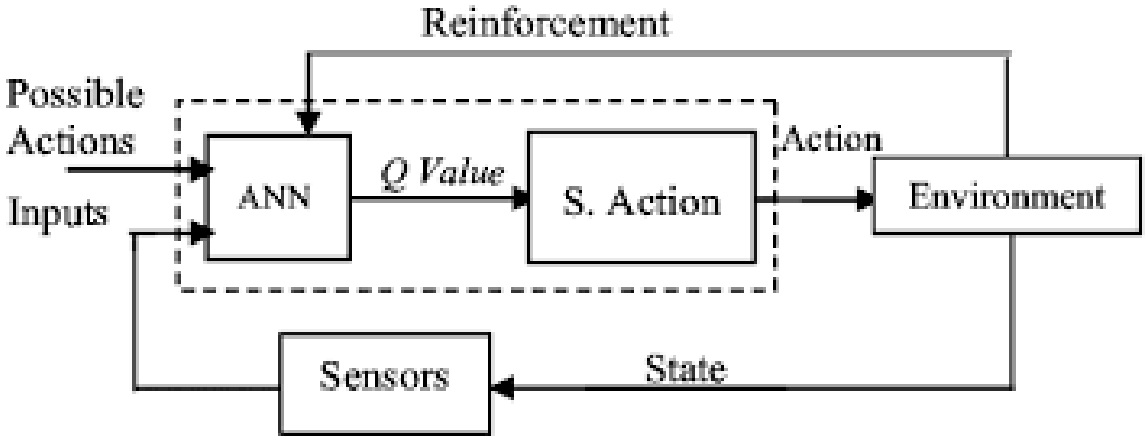
¿Cómo se hace una simulación? Es muy costoso explorar el espacio de estados

- El problema es estimar de manera eficiente la utilidad a partir de ejemplos
- Problema típico de Machine Learning
- Como se implemente esto con redes neuronales artificiales?
- Deep Q learning (Watkins, 1989) Esquema de redes neuronales aplicado a RL
- En vez de trabajar con U(s) se usa Q(s,a): utilidad acumulada y descontada cuando el estado es s y la acción es a (si conozco Q puedo evaluar U sumando sobre a)
- Q se actualiza de acuerdo a

$$Q_{t+1}(s,a) \leftarrow Q_t(s,a) + \alpha \left[r(s,a,s') + \gamma \max_{a'} Q_t(s',a') - Q_t(s,a) \right]$$

Reinforcement learning

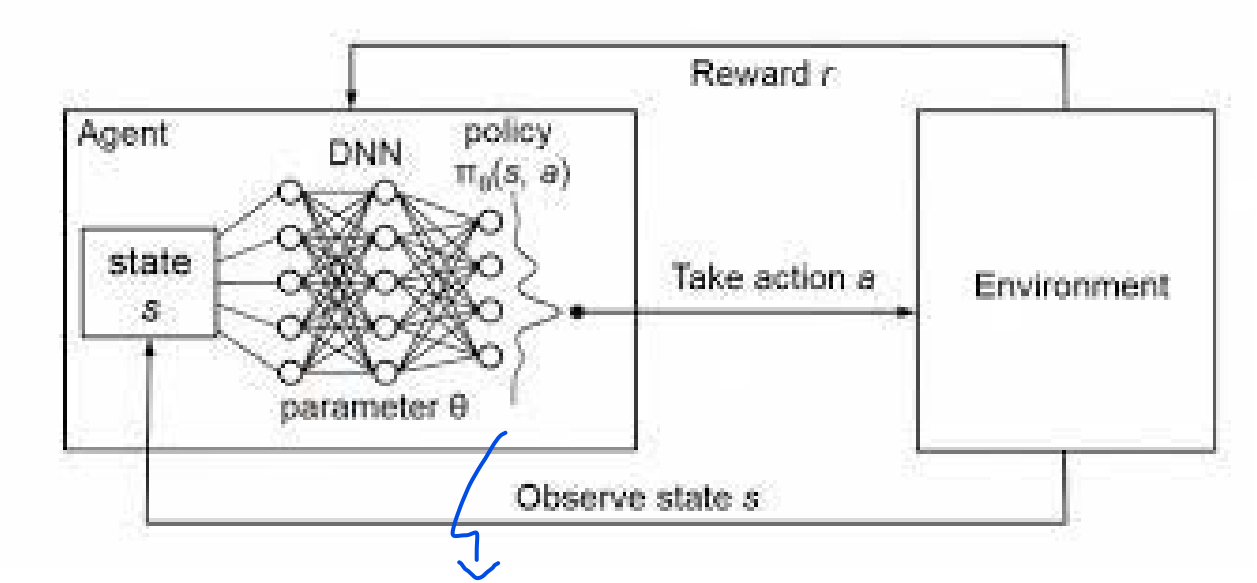
- Como se implementa esto con redes neuronales artificiales?
- Deep Q learning (Watkins, 1989)



Miden el entorno y dicen en qué estado está. Luego generan la entrada a la red neuronal

Reinforcement learning

- Como se implementa esto con redes neuronales artificiales?
- Deep Q learning (Watkins, 1989)



Cada unidad me dice la probabilidad de tomar una acción dada

Reinforcement learning

Resumen de ecuaciones:

- Como se implementa esto con redes neuronales artificiales?
- Supongamos que en el paso t del aprendizaje, la experiencia (s, a, s') fue muestreada.
- Quiero encontrar un "punto fijo"de $Q(s, a; \theta_t)$

$$Q_{t+1}(s,a) \leftarrow Q_t(s,a) + \alpha \left[r(s,a,s') + \gamma \max_{a'} Q_t(s',a') - Q_t(s,a) \right]$$

- Consideremos las variables

  $y = r(s, a, s') + \gamma \max_{a'} Q(s', a'; \theta_t)$

  $y\hat{} = Q(s, a; \theta_t)$

Reinforcement learning

- Como aprendizaje se puede intentar utilizar

  $\theta_t \rightarrow \theta_t - \alpha \nabla_\theta L$

  donde $L = (y - \hat{y})^2$

  Se mapea el problema a la minimización de una función de costo. Se mapea a backpropagation porque es lo que sabemos hacer

- Esto se parece a aprendizaje supervisado “normal”, pero tanto y como y^ se ajustan dinámicamente
- En la práctica esto lleva a over-fitting e inestabilidad. Entonces, se hace un batch de simulaciones
- Es necesario “separar” las escalas de tiempo, acumulando datos en un “buffer” D_t

En la práctica ya está programado esto

Reinforcement learning

- Ahora la función de loss es:

$$L(\theta) = \mathbb{E}_{(s,a,s')\sim U(D_t)}[(y - Q(s,a))^2] \approx \frac{1}{N}\sum_{n=1}^{N}(y_i - Q(s_i,a_i))^2$$

- Donde $U(D_t)$ significa que los datos son extraídos con probabilidad uniforme del buffer D_t
- Los valores de y_i son calculados usando una función fija $\underline{Q}$^, parametrizada por θ^
- Luego de usar el buffer se hace Q→ Q^

Reinforcement learning

- *Deep Q-learning*: se utiliza una red neuronal profunda para parametrizar Q: