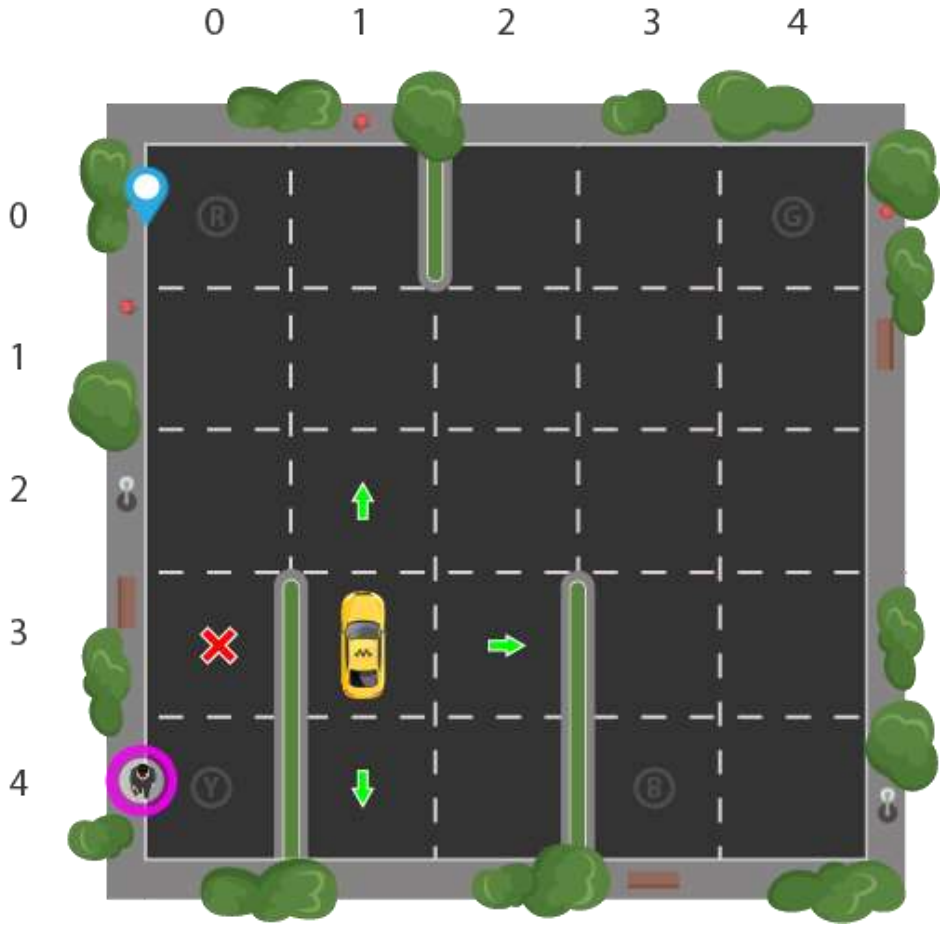
Q-Learning

DATA SCIENCE

Volviendo a nuestro SmartCab…



- Fue capaz de resolver el escenario de la figura…

… en unos discretos cientos de movimientos

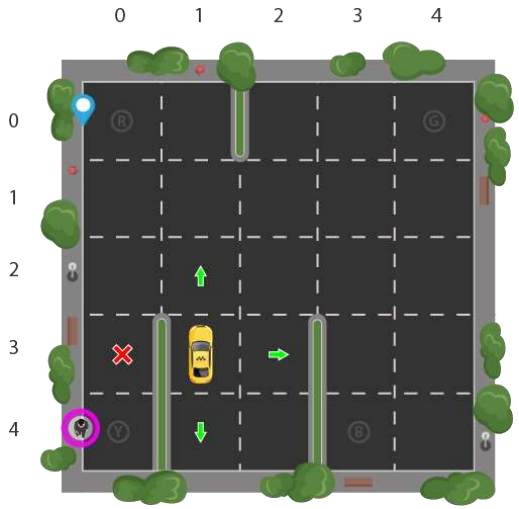
- Lo “aprendido” solo le sirve para ese escenario
- ¿Existe alguna manera de que aprenda realmente?

…imagina que tenemos una tabla…

- Una tabla que tuviera tantas filas como estados y tantas columnas como acciones posibles

Estado	Sur	Norte	Este	Oeste	Recoger	Dejar
0	…	…	…	…	…	…
…	…	…	…	…	…	…
328	-4	-1.8	-2.3	-3.8	-12	-12
…	…	…	…	…	…	…
499	…	…	…	…	…	…

- Cada celda indicara el valor de la mayor recompensa ACUMULABLE que podrías obtener si estando en el estado indicado por la fila ejecutaras la acción indicada por la columna



Estado: 328

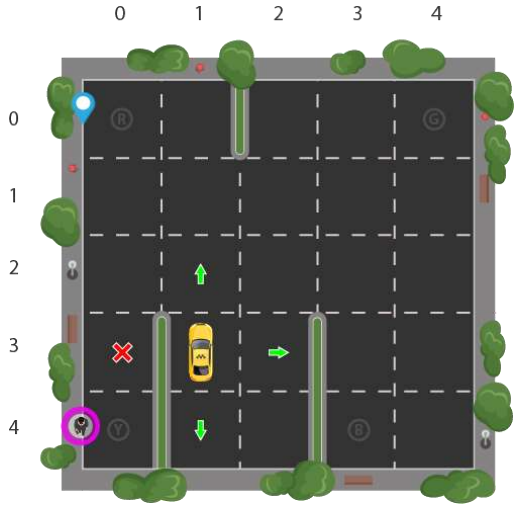
…imagina que tenemos una tabla…

- Una tabla que tuviera tantas filas como estados y tantas columnas como acciones posibles

Estado	Sur	Norte	Este	Oeste	Recoger	Dejar
0	…	…	…	…	…	…
…	…	…	…	…	…	…
328	-4	-1.8	-2.3	-3.8	-12	-12
…	…	…	…	…	…	…
499	…	…	…	…	…	…

- Cada celda indicara el valor de la mayor recompensa ACUMULABLE que podrías obtener si estando en el estado indicado por la fila ejecutaras la acción indicada por la columna

- Q-Learning: Dada esta tabla, el agente escoge siempre la acción con el valor máximo de su celda



Estado: 328

…una Q-Table (o tabla de valores Q)

Estado	Sur	Norte	Este	Oeste	Recoger	Dejar
0	…	…	…	…	…	…
…	…	…	…	…	…	…
328	-4	-1.8	-2.3	-3.8	-12	-12
…	…	…	…	…	…	…
499	…	…	…	…	…	…

- La Q-Table nos da para cada combinación (estado,acción) lo que se denomina su valor Q
- Los valores Q se representan como Q(s,a) donde s es el estado y a la acción
- En el ejemplo el Q(328,”sur”) es -4

¿Cómo se obtiene la tabla-Q?

- Inicializamos la tabla-Q intermedia con valores nulos
- El agente explorará de forma más o menos aleatoria el entorno
- Obteniendo valores de recompensa y actualizando una tabla-Q intermedia
- Continúa haciéndolo hasta alcanzar un criterio de parada

El agente “explora” el entorno

- Puede hacerlo de forma completamente aleatoria o

EXPLORATION 

- … combinando esta aleatoriedad con seguir la tabla-Q intermedia

EXPLOTATION 

- Épsilon (ε) es el hiperparámetro que regula la dicotomía Exploration vs Explotation

- ε -greedy

Actualización de la tabla-Q

$$Q(s, a) = (1 - \alpha) * Q(s, a) + \alpha * (r + \gamma * \max(Q(s', a')))$$

- Q(s, a) representa el valor Q para el estado s y la acción a.
- α (alfa) es la tasa de aprendizaje.
- r es la recompensa obtenida al tomar la acción a en el estado s.
- s' representa el nuevo estado resultante
- max(Q(s', a')) es el valor máximo de Q para el nuevo estado s' considerando todas las acciones posibles a'.
- γ (gamma) es el factor de descuento.

Algoritmo

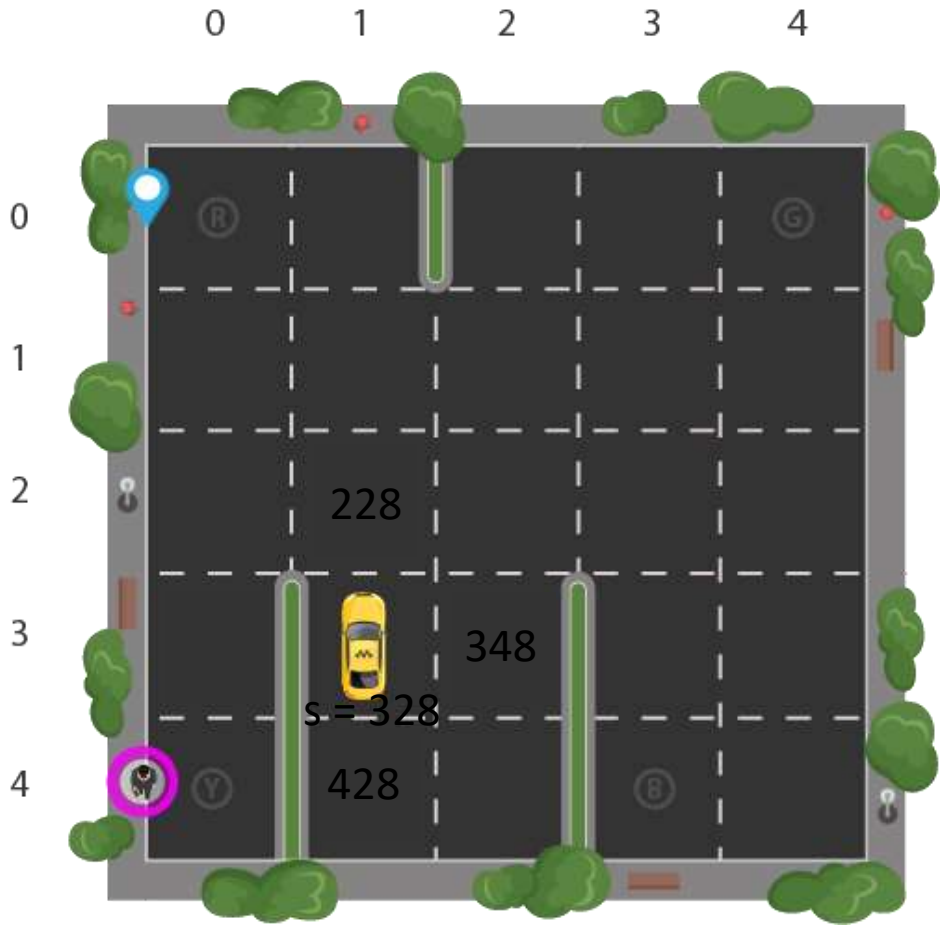
1. Inicializamos la tabla Q

1. Dar valores a los hiperparámetros. Valores de referencia:
   $\alpha = 0.05$; $\gamma = 0.9$; $\varepsilon = 0.1$

1. Ejecutar esta secuencia:
   I. Comprobar la condición de parada, si no se cumple ir a II
   II. Observar el estado actual (s)
   III. Escoger una acción aleatoria (explorar) o la acción con mayor Q(s,a) actual (explotar), en función de épsilon
   IV. Ejecutar la acción
   V. Actualizar Q(s,a) según la ecuación
   VI. Volver a I

Q-Learning a “mano” (I)

- ¿Te has perdido? Vamos a hacer un par de iteraciones “a mano” para que veas como funciona el algoritmo



- Partimos del estado 328 (s)
- Inicializamos la Q-table a cero

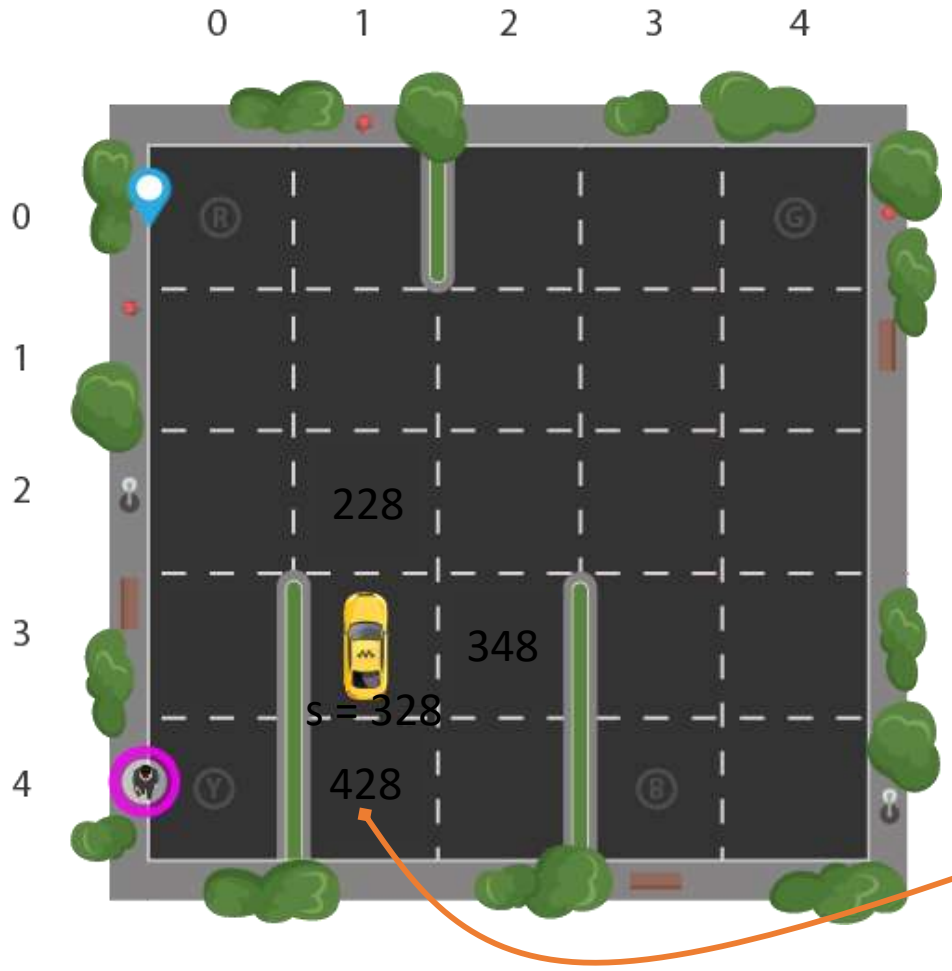
Estado	Sur	Norte	Este	Oeste	Recoger	Dejar
…	0	0	0	0	0	0
328	0	0	0	0	0	0
329	0	0	0	0	0	0
…	0	0	0	0	0	0

- Elegimos hiperparámetros:

$\alpha = 0.9$
$\gamma = 0.99$
Epsilon = 0.1

Q-Learning a “mano”

- ¿Te has perdido? Vamos a hacer un par de iteraciones “a mano” para que veas como funciona el algoritmo



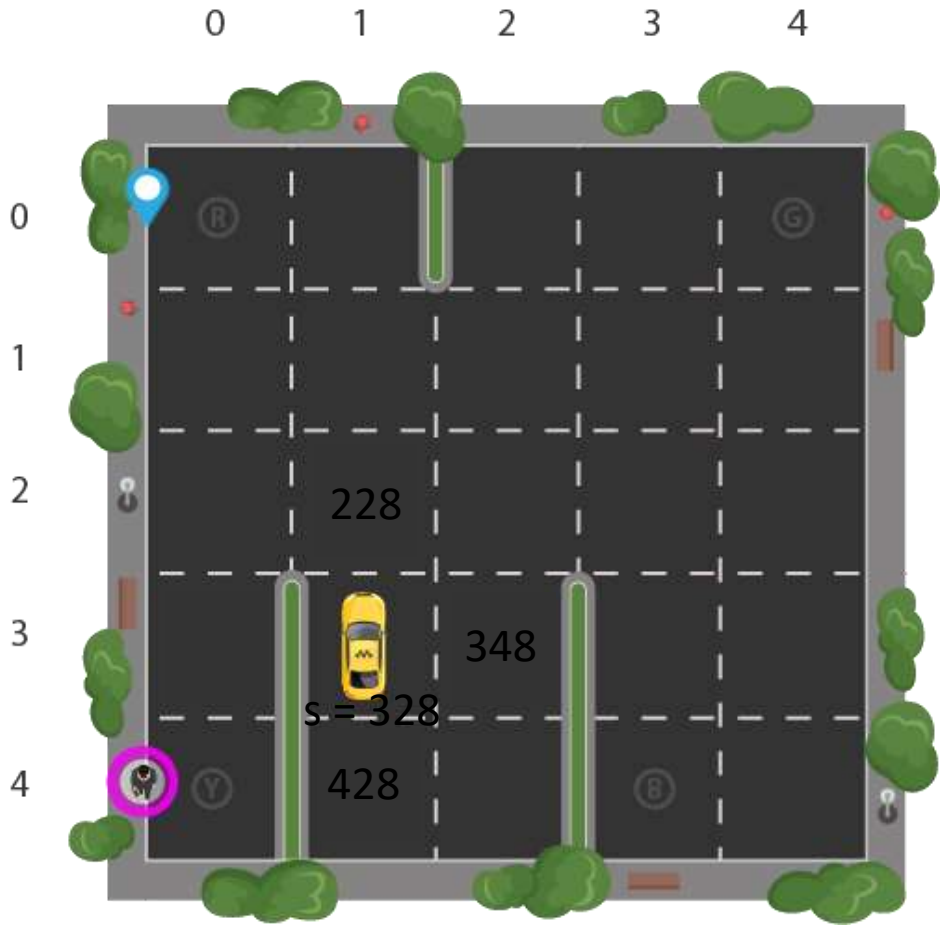
- Arrancamos el episodio
- Lanzamos un aleatorio entre (0,1) si el valor nos da menor que 0.1 elegiremos al azar si no la acción para nuestro estado como mayor Q (voz: Como estamos al principio ambas opciones son equivalentes)

Estado	Sur	Norte	Este	Oeste	Recoger	Dejar
…	0	0	0	0	0	0
328	0	0	0	0	0	0
329	0	0	0	0	0	0
…	0	0	0	0	0	0

- Nos sale acción: “sur”
- El agente pregunta al entorno y este nos dice:
  - s’ = 428 (el estado futuro)
  - r = -1 (la recompensa)

Q-Learning a “mano”

- ¿Te has perdido? Vamos a hacer un par de iteraciones “a mano” para que veas como funciona el algoritmo



- Actualizamos Q(328,sur) (ojo)

$$Q(328, sur) = (1 - 0.9) * Q(328,sur) + 0.9 * (-1 + 0.99 * \max(Q(428, a')))$$

- Observa que max(Q(428,a’) = 0,(voz: ya que todos los valores de Q están inicializados a 0 y no hemos actualizado ninguno todavía

$$Q(328\ sur) = 0 + 0.9 * (-1 + 0.99 * 0) = -0.9$$

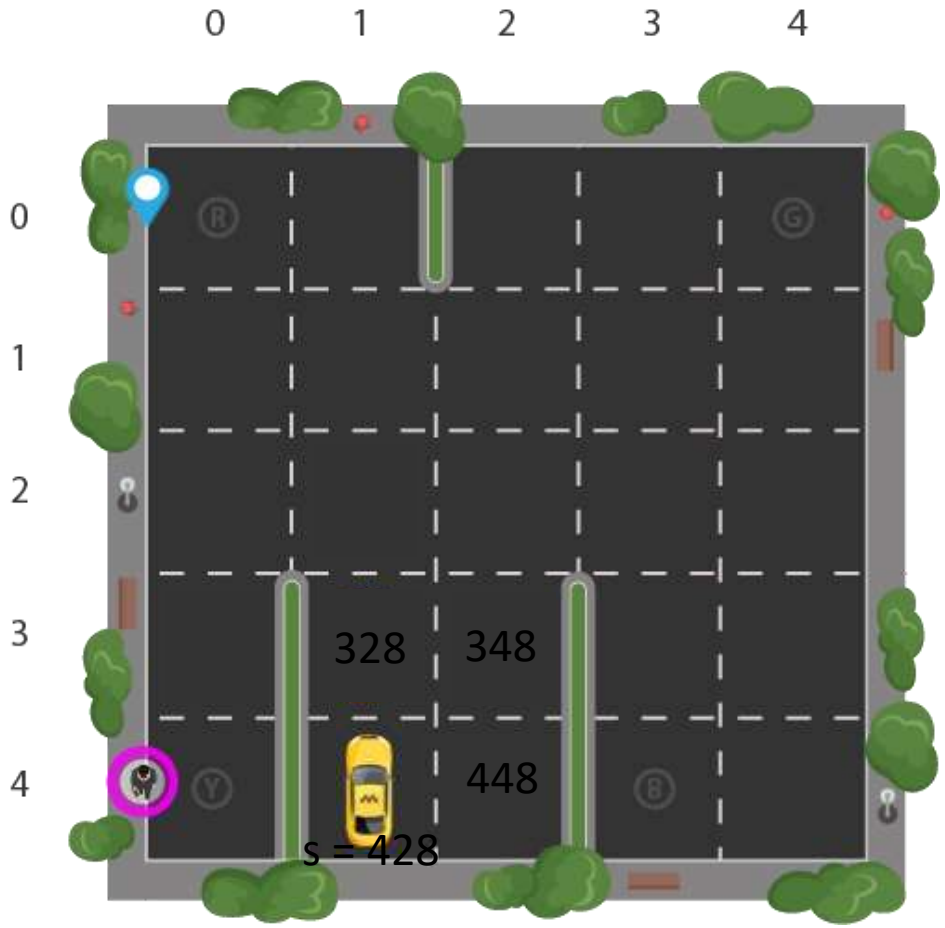
- Actualizamos la tabla

Estado	Sur	Norte	Este	Oeste	Recoger	Dejar
…	0	0	0	0	0	0
328	-0.9	0	0	0	0	0
329	0	0	0	0	0	0
…	0	0	0	0	0	0

- Movemos el coche y continuamos

Q-Learning a “mano”

- ¿Te has perdido? Vamos a hacer un par de iteraciones “a mano” para que veas como funciona el algoritmo



- Actualizamos Q(328,sur) (ojo)

$$Q(328, sur) = (1 - 0.9) * Q(328,sur) + 0.9 * (-1 + 0.99 * \max(Q(428, a')))$$

- Observa que max(Q(428,a’) = 0,(voz: ya que todos los valores de Q están inicializados a 0 y no hemos actualizado ninguno todavía

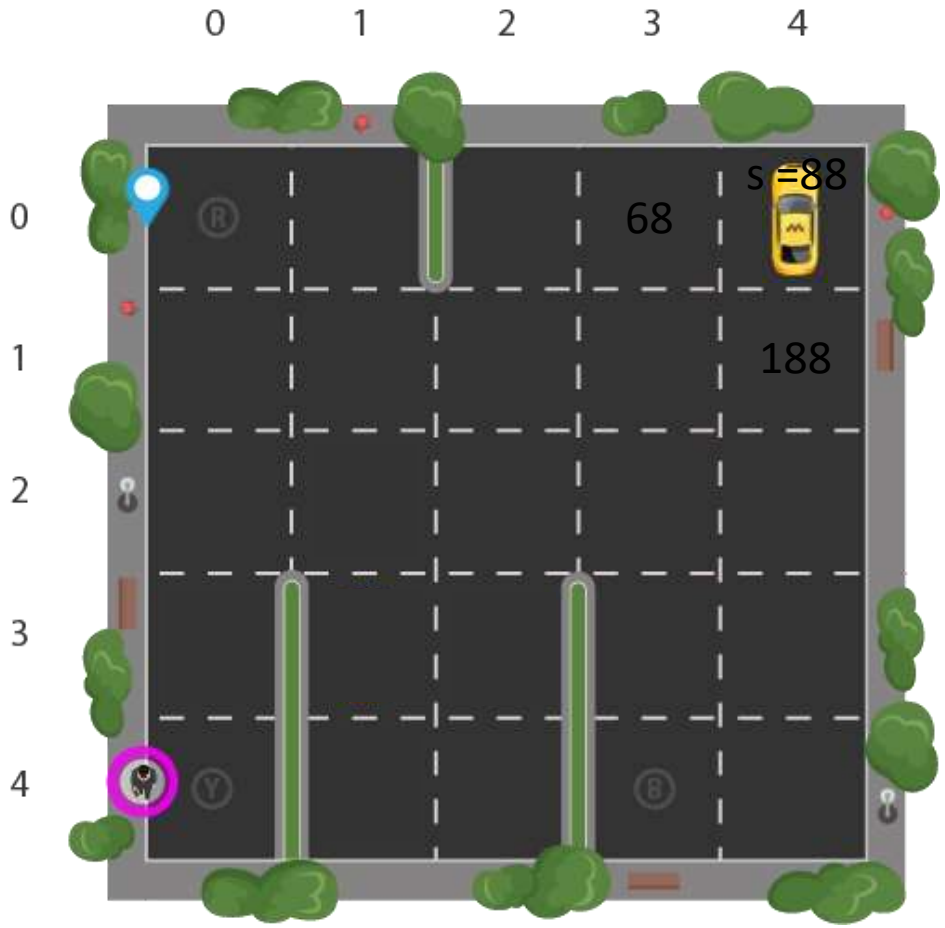
$$Q(328\ sur) = 0 + 0.9 * (-1 + 0.99 * 0) = -0.9$$

- Actualizamos la tabla

Estado	Sur	Norte	Este	Oeste	Recoger	Dejar
…	0	0	0	0	0	0
328	-0.9	0	0	0	0	0
329	0	0	0	0	0	0
…	0	0	0	0	0	0

- Movemos el coche y continuamos

Q-Learning a “mano”



- Avancemos unos cuantos pasos el “entrenamiento” y supongamos que estamos en el estado 88 con la siguiente Q-table

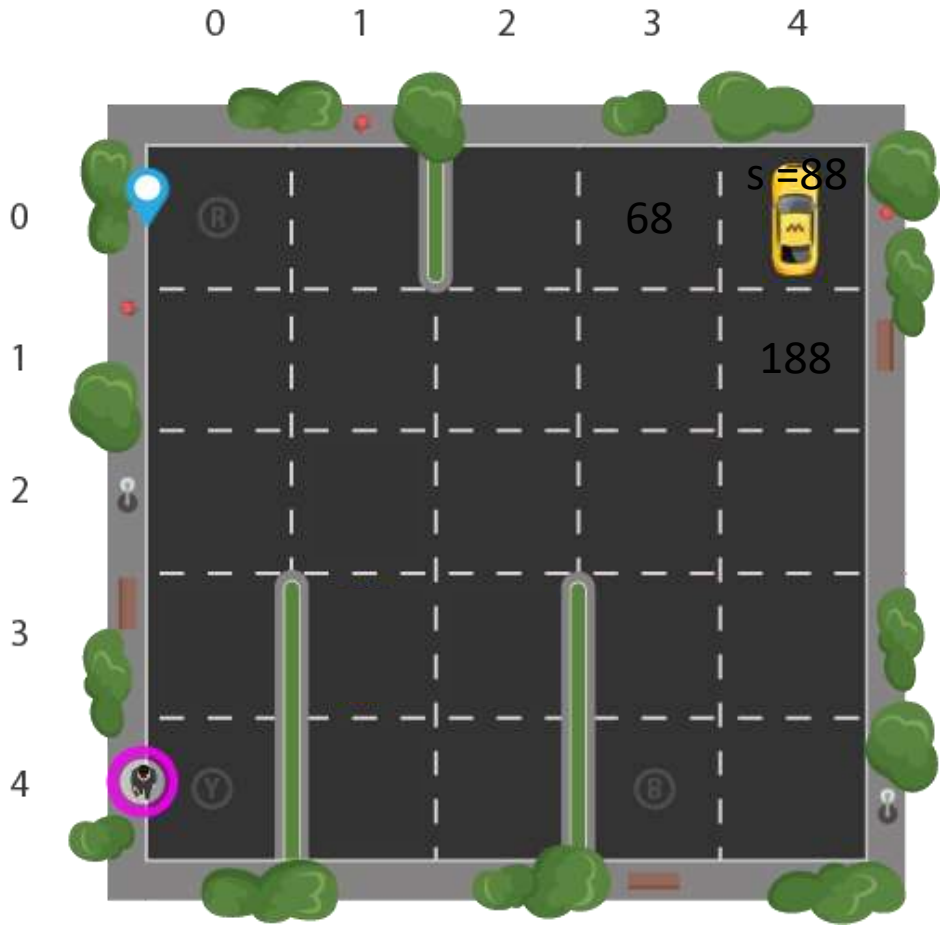
Estado	Sur	Norte	Este	Oeste	Recoger	Dejar
68	-2.3	-6.7	-4.3	-2.6	-10.12	-8.88
…	…	…	…	…	…	…
88	6	-3.3	-5.6	-2.2	12	16
….	…	…	…	…	…	…
188	-3.1	-4.2	-2.3	-5.5	-9.3	-7.68

- Lanzamos un aleatorio y nos sale por ejemplo 0.4 > épsilon, en vez de explorar haremos la acción de mayor Q, es decir “Dejar”

- El entorno nos devuelve (por ejemplo): s’ = 88 y r = -10 (voz: hemos querido dejar al pasajero y no está en el taxi, como corresponde al estado 88)

- Ahora a actualizar la tabla

Q-Learning a “mano”



- Actualizamos Q(88,Dejar) (ojo)

Q(88, Dejar) = (1 – 0.9) * 16 + 0.9 * (-10 + 0.99 * max(Q(88, a')))

- max(Q(88,a’), porque no hemos cambiado de estado con la acción Dejar

Q(88 , Dejar) = 1.6 + 0.9 * (-10 + 0.99 * 16) = 6.856

- Actualizamos la tabla (vaya cambio)

Estado	Sur	Norte	Este	Oeste	Recoger	Dejar
…	…	…	…	…	…	…
88	6	-3.3	-5.6	-2.2	12	6.856
…	…	…	…	…	…	…

De 16 a 6.856

Voz: Así es como rápidamente aprenderá que en este estado no debe dejar al pasajero desde el estado 88)