UT.7 Modelos basados en aprendizaje por refuerzo.

IES Abastos

CE Inteligencia Artificial y Big Data
Modelos de Inteligencia Artificial

Bloques de la UT7:

1. Introducción.
2. Visión de conjunto de los aprendizajes automático.
3. Agente y su entorno.
4. Tipos de aprendizaje por refuerzo.
5. Ejemplo 3 en raya.
6. Lab 1: Introducción a DeepRacer.
7. Lab 2: Carrera de clasificación para el campeonato nacional de AWS DeepRacer.

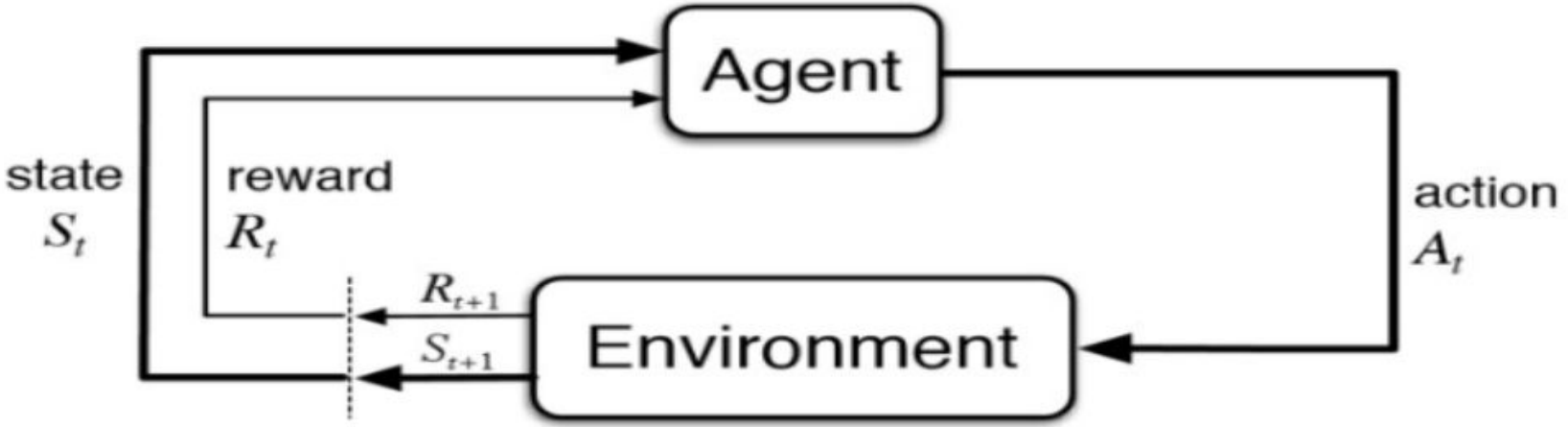
1. Introducción.

Los humanos aprendemos a realizar acciones en función del feedback obtenido. Las técnicas de Aprendizaje por Refuerzo están basadas en la tecnología conductista.

Objetivo: Establecer acciones que deben de ser elegidas en los diferentes estados con el objetivo de maximizar la recompensa.

1. Introducción.

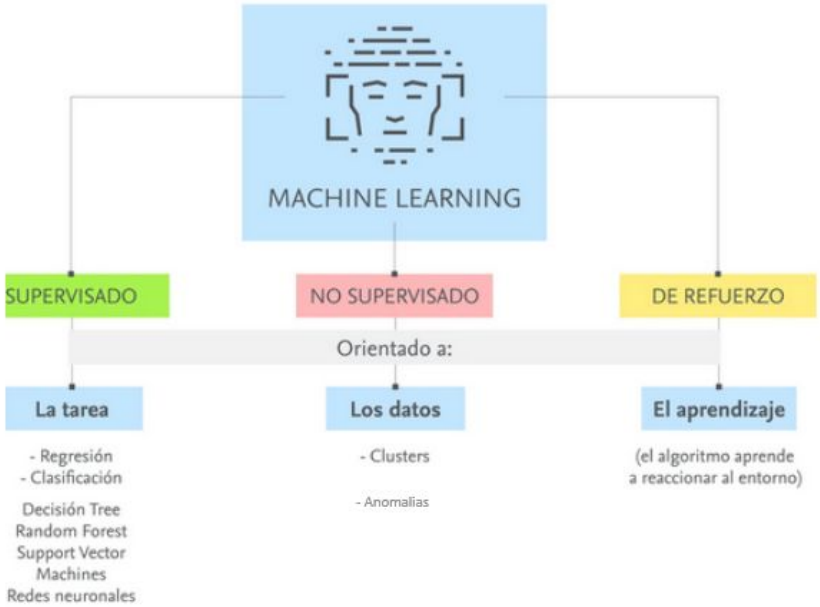
Ejemplo: Experimento Pavlov



1. Introducción.

Ejemplo: Procesado de datos.

2. Visión conjuntos de los tipos de aprendizaje.



3. Agente y su entorno.

Hay situaciones en las cuales el agente puede observar el entorno completo y son definidas como “**plena observabilidad**”.

En otras se trata de “**observabilidad parcial**”.

4. Tipos de aprendizaje por refuerzo.

Fuerza Bruta: Dos fases.

1. Para cada acción posible, muestrear los resultados.
2. Elegir la acción con el mayor retorno esperado.

¿ Problemas que pueden ocurrir ?

4. Tipos de aprendizaje por refuerzo.

Fuerza Bruta:

- El problema de este método es que el número de políticas suele ser extremadamente grande o incluso infinito.

- Además la varianza de los rendimientos puede ser muy grande, lo cual hace necesario un gran número de muestras para estimar con más precisión.

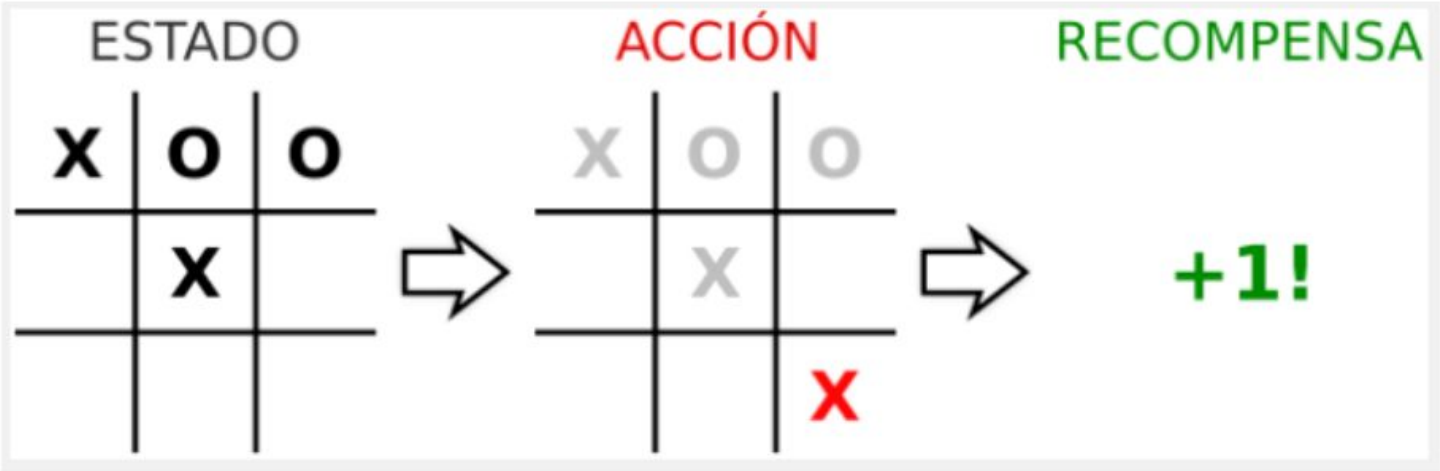
4. Tipos de aprendizaje por refuerzo.

Q-Learning: Algoritmo de Aprendizaje por refuerzo clásico, inventado hace más de 25 años.

- El agente aprende a asignar valores de bondad a los pares (estado, acción).
- Si un agente está en un estado y toma una determinada acción, estamos interesados en conocer el estado de esa acción, pero también de las recompensa futura (reward) de las posibles acciones posteriores.

5. Ejemplo 3 en raya.

Refuerzo positivo cada vez que el algoritmo gana. Probamos a hacer movimientos y observar el reward que proporcionan.



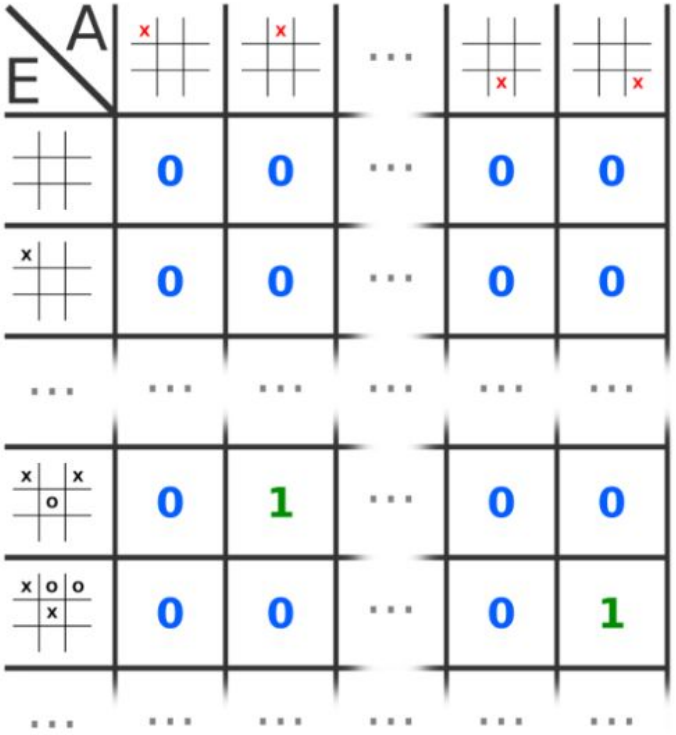
5. Ejemplo 3 en raya.

Tabla de recompensas: ¿Cual es la mejor jugada para cada estado?

- Filas=Estados
- Columnas=Acciones.

Cada celda contiene la recompensa recibida para cada acción/estado.

5. Ejemplo 3 en raya.



5. Ejemplo 3 en raya.

La mejor acción es la que tiene mayor recompensa (reward).
Solo conocemos las recompensas recibidas cuando ganamos la partida.

¿ Qué pasa en las acciones intermedias ?

Rellenar la tabla con las jugadas intermedias es el objetivo del algorimo Q-Learning.

5. Ejemplo 3 en raya.

¿ Qué hacer con las jugadas intermedias ?
¿Cual es la mejor jugada para cada estado?

La que tenga mayor recompensa a largo plazo.

Tabla: recompensas directas + largo plazo
Ejemplo. Ofrecer también recompensas cuando se empata y cuando se pierde

5. Ejemplo 3 en raya.

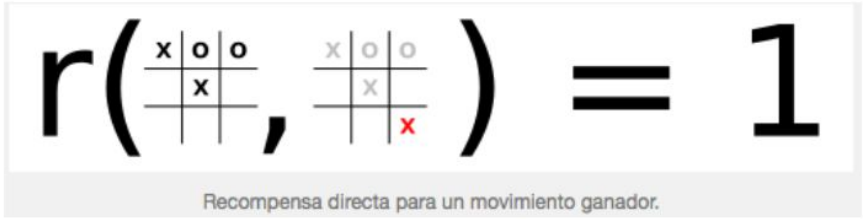
Recompensa a Largo Plazo y Mixta.

- La recompensa a largo plazo es la que esperamos obtener si en cada estado realizamos la mejor acción posible.

- La recompensa mixta es una combinación de la recompensa a largo plazo con la directa y se calcula de forma greedy.

5. Ejemplo 3 en raya.

Empiezo desde el final hasta el estado en que estoy.



5. Ejemplo 3 en raya.

Estado acción.

- Para un estado y una acción es posible tener múltiples experiencias (episodes).
- En el 3 en raya las experiencias dependen de lo que hace el rival.

5. Ejemplo 3 en raya.

Algoritmo.

- Hay que almacenar en memoria una tabla con las recompensas para estados y acciones.
- La tabla tiene la mejor estimación para la recompensa mixta.
- Al principio será mala pero irá aprendiendo.

5. Ejemplo 3 en raya.

Algoritmo parámetros: Velocidad de Aprendizaje: (learning rate):

Una velocidad de aprendizaje demasiado pequeña puede hacer que el modelo tarde mucho en converger o quede atrapado en mínimos locales, mientras que una velocidad de aprendizaje demasiado grande puede hacer que el modelo oscile y no converja.

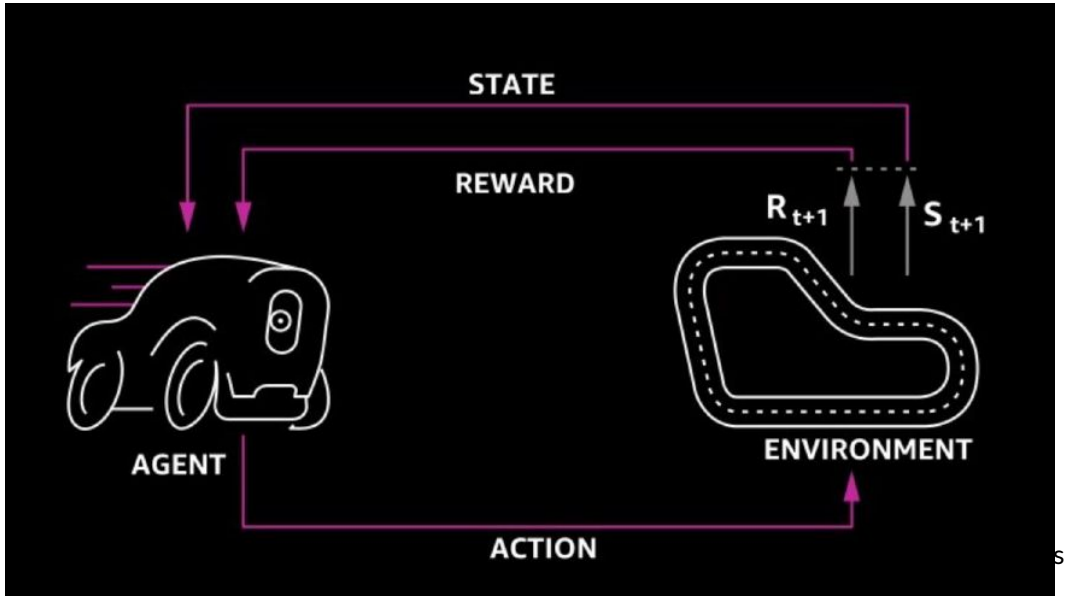
5. Ejemplo 3 en raya.

Algoritmo parámetros: Factor de descuento: (discount factor)

- Representa la preferencia temporal del agente. Un valor más cercano a 1 indica un agente que valora más las recompensas a largo plazo, mientras que un valor cercano a 0 indica un enfoque más inmediato.
- En el contexto de la función de valor en aprendizaje por refuerzo, se utiliza para descontar las recompensas futuras y determinar la importancia relativa de las recompensas inmediatas frente a las futura

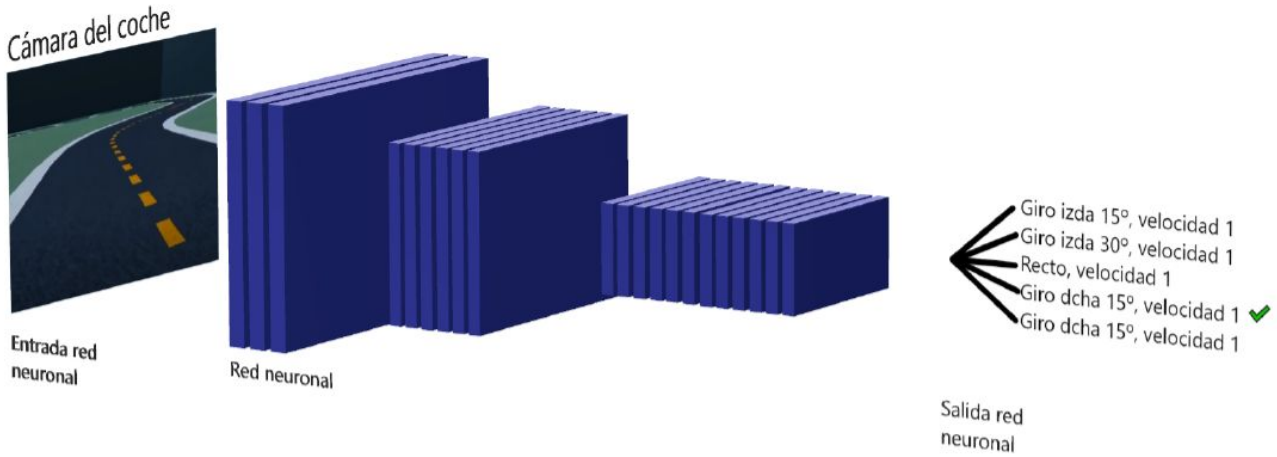
6. Introducción a DeepRacer.

Deep Racer: Es un coche autónomo a escala 1:18, que conduce gracias al Aprendizaje por Refuerzo.



6. Introducción a DeepRacer.

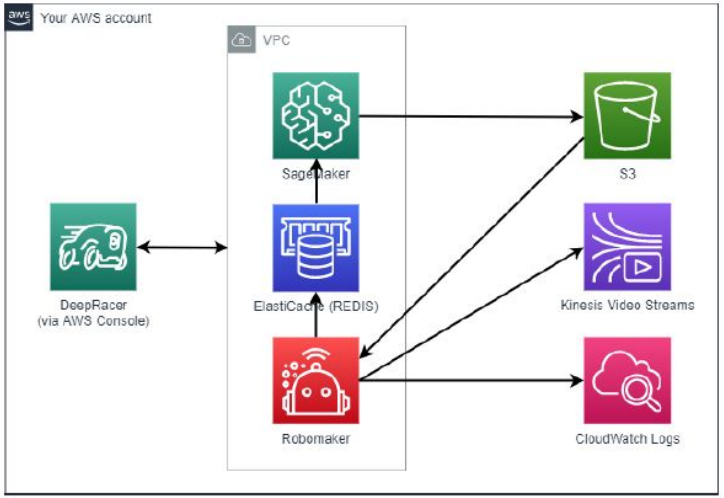
Curiosidad: La cámara es una red neuronal (Aprendizaje supervisado). Combinamos modelos y tipos de aprendizaje.



6. Introducción a DeepRacer.

Curiosidad DeepRacer utiliza internamente otros servicios de AWS:

- **S3**, para almacenar:
  - Hiperparámetros del modelo.
  - La función de recompensa.
  - Fichero de acciones posibles del coche.
  - Checkpoints del modelo.
- **SageMaker**, para entrenamiento.
- **RoboMaker**, para simulador.
- **Kinesis**, para vídeo streaming del simulador.
- **ElastiCache**, para comunicación de RoboMaker hacia SageMaker.
- **CloudWatch**, para logs.



6. Introducción a DeepRacer.

Los coches tienen sensores y mecanismos. Los sensores para saber el **contexto,** los mecanismo para ejecutar la **decisión**, en base a un **sistema cognitivo y** todo lo relevante se quedará almacenado **en un sistema bigdata:**

- Qué sensores usar (1 cámara, 2 cámaras, 2 cámaras + LiDAR).
- Ángulo de giro máximo de las ruedas delanteras.
- Velocidad máxima.
- Variación discreta(por pasos) o continuo.
- Si es por pasos, nºde pasos en el ángulo de giro de rueda.
- Si es por pasos, nºde pasos en la velocidad.

6. Introducción a DeepRacer.

Función de recompensa:

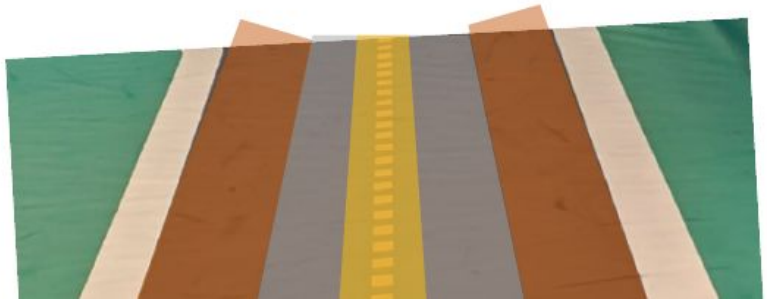
La función de recompensa no le dice al coche lo que tiene que hacer en cada instante; le premia de forma proporcional a lo bien que lo ha hecho en cada instante, durante el entrenamiento.

6. Introducción a DeepRacer.

Trucos:

Más fácil conseguir resultados con una función de recompensa sencilla.
El modelo aprende más rápido con funciones de recompensa continuas.

Función discontinua



Función continua

6. Introducción a DeepRacer.

Trucos:

El modelo aprende más rápido con coches “limitados”:

- Aprende más rápido el coche que solo puede ir a 1m/s, que un coche de 3-5 velocidades. Ídem para giros de volante (mejor 3 posiciones que 5, etc).
- Pero al clonar un modelo entrenado no podemos cambiar de coche.
- Idea: empezar con un coche que tenga muchas velocidades pero limitarlas en la función de recompensa.

6. Introducción a DeepRacer.

Trucos:

Empezar con una función de recompensa muy sencilla, hasta conseguir que el coche complete la vuelta. Después clonar el modelo y aplicar “extra” en la función de recompensa.

•Ejemplo:

•1º Recompensar solo por ir centrado en la pista pero con velocidad fija.

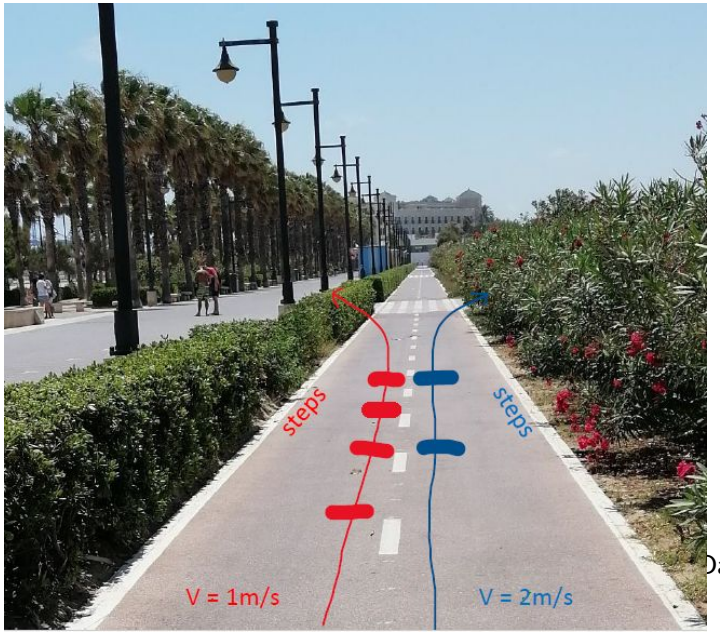
•2º Clonar modelo.

•3º Entrenar el modelo clonado, recompensando también por progresar (≠ ir rápido).

6. Introducción a DeepRacer.

Trucos:

- Es más importante el “progreso” que la “velocidad”.



6. Introducción a DeepRacer.

Gracias por los trucos a:

Presentación

Javier Campos (@javichur)

AWS Faculty Lead en CEU Digital

Cofundador de Mobilendo (2010-2020)

+10 años creando **apps** (Dev, PM, CTO)

Alexa Champion

#LifelongLearning #OpenSource
#Serverless #Hackathon #ML #NLP

6. Introducción a DeepRacer.

Ahora nos descargamos el PDF de la actividad en AWS Academy.