Anexo: Marco teórico

El aprendizaje automático se puede definir como un campo dentro de la inteligencia artificial. Este campo estudia la generación de algoritmos y programas que aprenden. Es decir, cambian cuando son expuestos a nuevos datos. En el aprendizaje automático, se detectan patrones en los datos proporcionados para ajustar correctamente las acciones que el programa ha de realizar. Dentro del aprendizaje automático, podemos distinguir dos tipos: aprendizaje supervisado y no supervisado. El aprendizaje supervisado es capaz de aplicar lo que se ha aprendido en el pasado a nuevos datos, mientras que el no supervisado puede extraer inferencias de un conjunto de datos. [1]

En el aprendizaje por refuerzo, nuestro algoritmo recibirá cierta valoración a cerca de la idoneidad de la respuesta dada. Si la respuesta es correcta, este aprendizaje es muy parecido al aprendizaje supervisado. Donde realmente difieren estos tipos de aprendizaje es en las respuestas erróneas. Entonces, el aprendizaje supervisado es capaz de decirle al aprendiz exactamente qué es lo que debería haber respondido. El aprendizaje por refuerzo, por su lado, solo le informa de que el comportamiento no ha sido el adecuado. [2]

El aprendizaje por refuerzo tiene comparación en la vida real con los aprendizajes de algunos animales. Por ejemplo, cuando a un perro se le da una orden, ya sea sentarse, tumbarse, etc. Si el perro realiza la orden, se le recompensa. En cambio, si no la realiza, recibe algún tipo de castigo. De esta forma, con repetición, podemos conseguir que el perro refuerce sus comportamientos que reciben recompensa, aprendiendo poco a poco. [2]

Este tema podría expandirse a, por ejemplo, un videojuego. Pondremos el ejemplo del famoso Pacman. En este videojuego, contamos con un agente (Pacman), un estado (la ubicación) y un medio ambiente (el laberinto). Pacman puede obtener distintos tipos de recompensas: positivas, si obtiene puntos por comer, y negativas, tras morir al cruzarse con un fantasma. Estas recompensas estarán definidas en base a las acciones que vaya realizando en todo momento. El algoritmo del aprendizaje por refuerzo, más allá de la recompensa inmediata, busca la recompensa acumulada. Esto hace que este tipo de aprendizaje se base en la repetición de un bucle de estado – acción – recompensa. [3]

Nos encontramos ahora con el algoritmo Q learning, en el que el agente aprende a asignar valores de bondad a los pares (estado, accion). En este algoritmo, un agente que se encuentre en un estado contempla la recompensa inmediata a obtener, pero también las futuras recompensas que obtendrá según la acción que decida tomar. De esta forma, el valor Q de un par (estado, acción) contiene la suma de todas las posibles recompensas en cada estado para cada acción realizable. [2]

Este algoritmo tiene ciertas variantes en su implementación, las cuales veremos ahora:

* Q-learning profundo: El sistema utiliza una red neuronal convolucional. La técnica que se utiliza es la experiencia de reproducción, un mecanismo de inspiración biológica que utiliza una muestra aleatoria de acciones previas en lugar de la acción más reciente para continuar. [4]
* Doble Q-learning: Dos funciones separadas se entrenan de forma simultánea utilizando experiencias separadas. Esto nos resuelve un problema de sobreestimación. Este algoritmo fue posteriormente combinado con el Q-learning profundo. [4]
* Q-learning retardado: Implementación alternativa que utiliza el aprendizaje PAC. En este tipo de aprendizaje, se reciben muestras y se selecciona una función de generalización de una clase segura de funciones posibles. [5]
* Greedy GQ: Utiliza q-learning en combinación con una función de aproximación.

Ahora hablaremos del algoritmo SARSA. Este algoritmo es muy parecido a Q-learning, simplemente cambia la función de actualización en la que se escoge la acción actual tomada para tenerla, en vez de la acción óptima. En este algoritmo, la función que actualiza Q queda así: rt + Gamma\*Qt(st+1,at+1) 🡪 Qt+1(st, at). A consecuencia de esta diferencia en la actualización, cuando necesitamos la acción actual para la actualización, tenemos que replantear el esquema para primero tomar una acción, y luego entrar en un bucle en el que se evaluarán estado y recompensa, tomar otra acción y actualizar. [6]

Referencias:

[1] Aprendizaje automático (machine learning): <https://searchdatacenter.techtarget.com/es/definicion/Aprendizaje-automatico-machine-learning>

[2] Aprendizaje por refuerzo: algoritmo Q learning: <http://www.cs.us.es/~fsancho/?e=109>

[3] Conceptos de inteligencia artificial: qué es el aprendizaje por refuerzo: <https://www.xataka.com/inteligencia-artificial/conceptos-inteligencia-artificial-que-aprendizaje-refuerzo>

[4] Q-learning: <https://es.qwe.wiki/wiki/Q-learning#Variants>

[5] Aprendizaje PAC: <https://es.wikipedia.org/wiki/Aprendizaje_PAC>

[6] Aplicación de técnicas de aprendizaje automático sobre juegos: <https://riunet.upv.es/bitstream/handle/10251/55343/AGUADO%20-%20Aplicaci%C3%B3n%20de%20t%C3%A9cnicas%20de%20aprendizaje%20autom%C3%A1tico%20sobre%20juegos.pdf?sequence=1>