

Investigación y Gestión de Proyectos en Inteligencia
Artificial

Tema 8. Investigación en agentes inteligentes y sistemas expertos

Índice

Esquema

Ideas clave

- 8.1. Introducción y objetivos
- 8.2. Introducción a los agentes inteligentes
- 8.3. Comportamiento y entorno de los agentes inteligentes
- 8.4. Estructura de los agentes inteligentes
- 8.5. Proyectos de investigación en agentes inteligentes
- 8.6. ¿Qué son los sistemas expertos?
- 8.7. Proyectos de investigación en sistemas expertos
- 8.8. Referencias bibliográficas

A fondo

- Mastering the game of Go with deep neural networks and tree search
- Expert Systems

Test



8.1. Introducción y objetivos

En este tema se explica a alto nivel el **concepto y la estructura general de los agentes inteligentes y los sistemas expertos**. Estos conceptos forman parte de los fundamentos básicos de la inteligencia artificial y, de una forma u otra, deben considerarse a la hora de diseñar una solución basada en esta disciplina.

De forma detallada, los **objetivos** que persigue esta unidad son:

- ▶ Ser capaz de definir las peculiaridades de un agente inteligente.
- ▶ Identificar las aplicaciones de los agentes inteligentes.
- ▶ Conocer los fundamentos de diseño y estructura de los agentes inteligentes.
- ▶ Conocer los conceptos clave de los sistemas expertos.
- ▶ Identificar qué proyectos son más adecuados para los sistemas expertos.

8.2. Introducción a los agentes inteligentes

En temas anteriores se ha discutido el significado de inteligencia y comportamiento racional. Todos estos contenidos previos deben tenerse en mente a la hora abordar el presente tema.

Un agente es un ente que recibe información del entorno y actúa en consecuencia. Se espera de él que actúe de forma inteligente.

Un ejemplo de actuación inteligente es la que está orientada a la **maximización de un objetivo** previamente establecido usando toda la información disponible en ese momento. Por ejemplo, si el objetivo es la supervivencia, una acción inteligente sería escapar u ocultarnos de un depredador.

De forma abstracta, un agente es una función que convierte las percepciones en acciones.

$f: P^* \rightarrow A$

P^* : indica un conjunto de percepciones finito.

A : indica la acción que llevamos a cabo en función de dichas percepciones.

Los agentes son muy útiles para modelar entornos complejos donde las acciones que lleven a cabo unos agentes puedan afectar a movimientos posteriores de otros entes. Se recibe información de un entorno y las propias acciones que los agentes llevan a cabo insertan nueva información en dicho entorno.

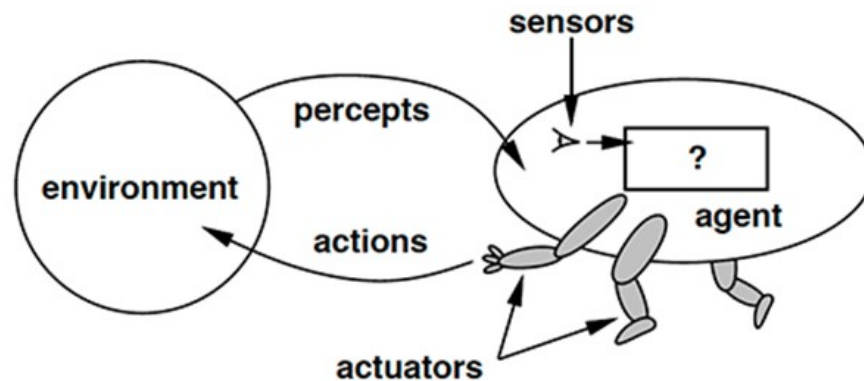


Figura 1. Un agente y su entorno. Fuente: Russell, 2004.

Ilustraremos estos conceptos mediante un sencillo ejemplo práctico. Supongamos que hemos creado un robot aspirador. Hemos hecho un gran trabajo y, por tanto, nuestro robot puede considerarse un buen ejemplo de agente inteligente. ¿Cómo actuaría este robot aspirador? Resumiendo, la función asociada al agente (que estaría ejecutándose de forma continua mientras el robot estuviese encendido) podría ser como la que sigue (Russell, 2004):

```
function Reflex-Vacuum-Agent([location,status]) returns an action
```

```
if status = Dirty then return Suck
```

```
else if location = Right then return Right
```

```
else if location = Left then return Left
```

Explicado de forma sencilla, el robot dispone de un sensor que, dentro de un radio de acción, le indica que hay algún tipo de residuo o suciedad por recoger. Si así fuese, el mismo sensor le proporciona la localización

concreta donde actuar. Si está a su derecha, el robot toma la decisión de moverse a la derecha. Una vez posicionado en el punto concreto, activa el aspirador. Tras ejecutar estas acciones, el entorno de partida ha cambiado, el residuo o resto de basura ha desaparecido y, por tanto, la información asociada a dicho entorno no es la misma que al inicio.

Como podemos esperar, sacar al mercado un robot aspirador implica crear un producto más completo del aquí reflejado. No obstante, la dinámica del procedimiento es común a todos los agentes inteligentes, todos tienen en cuenta la información que proporciona el entorno para, en función de una configuración previa, llevar a cabo una serie de acciones que permite maximizar el objetivo inicial planteado.

8.3. Comportamiento y entorno de los agentes inteligentes

Para diseñar un agente inteligente **debemos especificar de antemano el entorno en el que se va a ejecutar la tarea**. Esto implica definir cómo se va a evaluar si la tarea se está realizando correctamente, qué información se recoge del entorno, cuáles son las posibles acciones y qué tipo de sensores vamos a emplear. Todo ello constituye lo que se llama el **entorno de trabajo**, para cuya denominación se utiliza el acrónimo **REAS** (rendimiento, entorno, actuadores/acciones, sensores).

Por ejemplo, supongamos que queremos diseñar ahora un coche autónomo. Podríamos trabajar con la siguiente descripción de elementos:

- ▶ **Indicadores de rendimiento:** número de kilómetros recorridos, número de infracciones de tráfico cometidas, número de accidentes, valoración media (o mediana) de la comodidad del viaje por parte de los ocupantes del vehículo, número de situaciones de riesgo detectadas, coste del sistema, etc.
- ▶ **Entorno:** calles de la ciudad, señales, otros vehículos, peatones, etc.
- ▶ **Acciones posibles:** arrancar, parar, frenar, acelerar, girar, cambiar marcha, etc.
- ▶ **Sensores:** velocímetros, frenos, GPS, cámaras de vídeo, etc.

A la hora de diseñar el agente se debe tener en especial consideración las **características del entorno**. Por ejemplo, no es lo mismo un entorno totalmente observable y una lógica de secuencias simple como puede ser la planteada para aspirar una habitación, que un entorno más estocástico e incluso con fenómenos no

directamente observables como pueden ser los mercados financieros. En este último caso y en el ejemplo concreto del diseño de un agente inteligente para llevar a cabo operaciones bursátiles automatizadas, es probable que no se disponga del detalle de todas las acciones llevadas a cabo por todos los agentes. Podríamos llegar a percibir las consecuencias de alguna acción concreta sin tener totalmente claro a quién asociar la acción raíz.

Propiedades de los entornos de trabajo

Los agentes inteligentes se emplean en entornos muy variados y diversos. Aun así, es posible identificar un número de dimensiones concretas en las que categorizar estos entornos. Estas dimensiones influyen de manera crítica en el diseño del agente. Las dimensiones que podemos considerar son las siguientes:

Totalmente observable vs. parcialmente observable

Diremos que el entorno de trabajo es totalmente observable si los sensores del agente le proporcionan información sobre todos los aspectos relevantes para la toma de decisiones. En otro caso, el entorno de trabajo es parcialmente observable. Un entorno puede ser parcialmente observable debido al ruido y a la existencia de sensores poco exactos o porque los sensores no reciben información de parte del sistema. Por ejemplo, en un sistema de inversión bursátil automatizado hay que tomar decisiones sin tener visión de qué están plateando o ejecutando otros actores en ese mismo momento.

Determinista vs. estocástico

Un entorno de trabajo es determinista cuando el siguiente estado del entorno está totalmente determinado por el estado actual y la acción ejecutada por el agente. En otro caso, hablamos de un entorno estocástico. Un entorno estocástico añade incertidumbre a la toma de decisiones. Por ejemplo, y en el caso de uso de un coche autónomo, el entorno es claramente estocástico, ya que en cualquier momento

puede aparecer una avería, pinchazo o peatón inoportuno que altere el rumbo de los acontecimientos.

Secuencial vs. episódico

En un entorno secuencial las acciones realizadas en el corto plazo pueden afectar a largo plazo. Un ejemplo claro es el motor inteligente de un juego de ajedrez. Un mal movimiento puede dar al traste con una partida, aunque el jaque mate no llegue de manera inmediata. Por el contrario, en un entorno de trabajo episódico, las acciones a realizar en el siguiente episodio no dependen de las acciones que se realizaron en episodios previos. Por ejemplo, un robot ubicado en una cadena de montaje puede tener el objetivo de detectar piezas defectuosas y sacarlas del ciclo de fabricación. En este caso no se tiene en cuenta las piezas descartadas con anterioridad ni tampoco lo que se haga con esas piezas en el futuro. Los entornos episódicos son más simples puesto que no requieren el estudio de ciertas dependencias entre las acciones que se llevan a cabo.

Estático vs. dinámico

En un entorno de trabajo dinámico, el entorno puede cambiar mientras el agente está analizando la situación para decidir qué acción tomar. En otro caso, el entorno es estático. En el caso del coche autónomo, el entorno es dinámico puesto que todos los vehículos y peatones se están moviendo mientras el robot decide qué es lo próximo que debe hacer.

Discreto vs. continuo

Un entorno de trabajo discreto tiene un número finito de estados distintos, aunque este número sea muy elevado. Esta situación se da en juegos como el ajedrez, que poseen un elevadísimo número de combinaciones, pero aun así finito. Por el contrario, el recurrido caso del coche autónomo sería un ejemplo de entorno de trabajo continuo, puesto que la velocidad y ubicación de los coches y peatones

tienen un rango de valores continuos a los que hay que unir parámetros varios como ángulos de giro, inclinación de la carretera, etc.

Agente individual vs. multiagente

Un entorno de trabajo donde un único agente trata de resolver un determinado problema sin interactuar con otros entes inteligentes sería un ejemplo de entorno de trabajo con agente individual. Por el contrario, entornos de trabajo como el ajedrez, donde dos agentes compiten entre sí o como el caso de uso del coche autónomo, donde varios agentes inteligentes tienen que convivir, serían ejemplos de entorno de trabajo multiagente.

Encontraremos el caso más complejo ante un entorno de trabajo que sea parcialmente observable, estocástico, secuencial, dinámico, continuo y multiagente. Realmente, este caso extremo suele ser uno de los más frecuentes.

La tabla siguiente muestra la clasificación de algunos ejemplos de uso según la tipología del entorno de trabajo (Russell, 2004). Todos los casos se refieren a un agente inteligente realizando la labor referenciada, ya puede ser jugar a un determinado juego, conducir un taxi (robotizado) o enseñar inglés.

Entorno de trabajo	Observable	Determinista/estocástico	Secuencial/episódico	Estático/dinámico	Discreto/continuo	Individual/multiagente
Crucigrama	Totalmente	Determinista	Secuencial	Estático	Discreto	Individual
Ajedrez	Totalmente	Determinista	Secuencial	Semi	Discreto	Multi
Póker	Parcialmente	Determinista	Secuencial	Estático	Discreto	Multi
Backgammon	Totalmente	Estocástico	Secuencial	Estático	Discreto	Multi
Taxi autónomo	Parcialmente	Estocástico	Secuencial	Dinámico	Continuo	Multi
Diagnóstico médico	Parcialmente	Estocástico	Secuencial	Dinámico	Continuo	Individual
Análisis de imagen	Totalmente	Determinista	Episódico	Semi	Continuo	Individual
Robot clasificador	Parcialmente	Estocástico	Episódico	Dinámico	Continuo	Individual
Tutor interactivo de inglés	Parcialmente	Estocástico	Secuencial	Dinámico	Discreto	Multi

Tabla 1. Ejemplos de entorno de trabajo y sus características. Fuente: Russell, 2004, p. 50.

8.4. Estructura de los agentes inteligentes

Un agente se compone de **arquitectura más software**. Por arquitectura entendemos el conjunto de sensores físicos, unidades de procesamiento y todo tipo de dispositivo que permite al agente comunicarse con el entorno y desencadenar acciones. El *software* recoge las instrucciones que guían el comportamiento del agente.

Podemos generalizar los distintos tipos de agentes inteligentes en cuatro grandes categorías de complejidad creciente. Esto significa que agentes categorizados en capas de mayor complejidad podrían ser capaces de realizar tipologías de tareas asociadas a agentes más simples, pero no al revés.

En primer lugar, encontramos a los **agentes reactivos simples**. Estos agentes reaccionan ante un estímulo o información de entorno externo y actúan en función de un conjunto de reglas o instrucciones previamente estipuladas. Toda la información previa o histórica es descartada. Por ejemplo, supongamos que los sensores de un robot aspirador informan de que justo en la ubicación actual hay unas migas de pan. El robot pasaría a succionar la suciedad sin plantearse nada más.

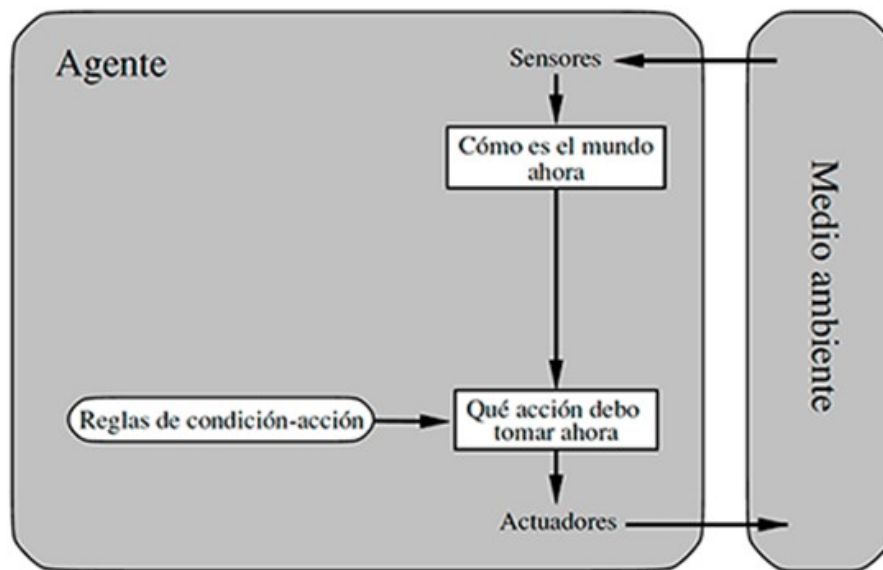


Figura 2. Agente reactivo simple. Fuente: Russel, 2004, p. 54.

Estos agentes se encuentran bastante limitados, puesto que solo se puede tomar la decisión correcta si el entorno es totalmente observable.

Una evolución más completa de estos agentes son los **agentes reactivos basados en modelos**. Estos agentes incorporan el histórico de información a la toma de decisiones. La historia previa se refleja en un determinado estado interno del agente, por lo que ya no solo se considera la información proveniente del entorno, sino que además debo tener en cuenta el estado en el que se encuentra el agente. Por ejemplo, imaginemos un videojuego de acción donde en un momento dado están disparando a nuestro personaje. El nivel de salud del personaje podría bajar o, por el contrario, podría mantenerse estático si nos encontramos en un estado «protegido» porque previamente hemos capturado algún tipo de objeto que ha creado un escudo protector. El estado del juego sería capaz de representar este amplio abanico de situaciones y entender cómo funciona su mundo particular. Y para entender su mundo necesita representar dicho conocimiento de manera formal, ya sea mediante un sistema de reglas, transacciones de estados, circuitos *booleanos* o ecuaciones matemáticas. Esa representación es lo que conocemos como modelo. Un agente que

utilice este tipo de modelos es un agente basado en modelos.

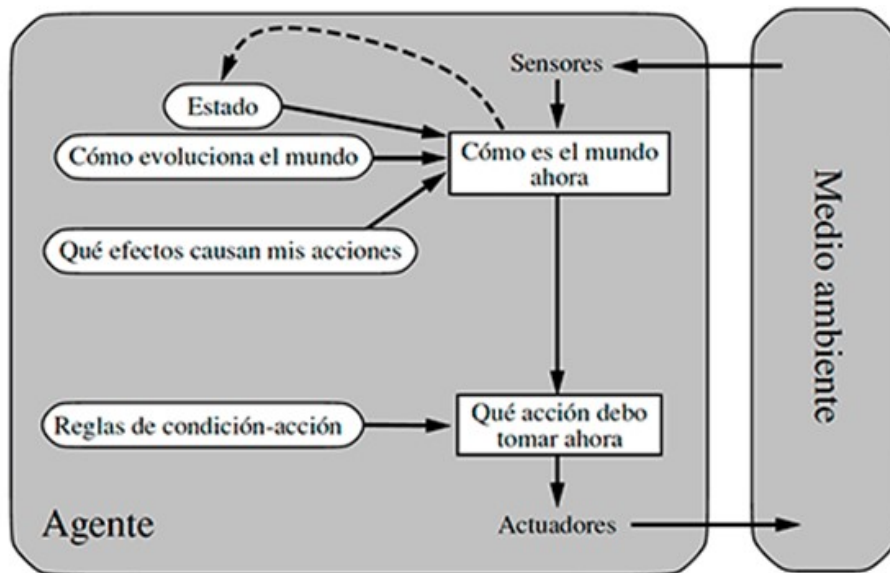


Figura 3. Agente reactivo basado en modelos. Fuente: Russell, 2004, p. 56.

Los **agentes inteligentes como los anteriores hacen que sea posible determinar de antemano un conjunto de reglas de condición-acción**. Por ejemplo, si los sensores del vehículo autónomo detectan que el coche de delante frena, se debe iniciar la frenada. Si los sensores del robot aspirador detectan suciedad a la derecha, el robot debe desplazarse a la derecha.

En muchas ocasiones no se dispone del conocimiento necesario para poder construir este conjunto de reglas, por lo que se establece un objetivo a conseguir. Un objetivo o meta podría ser, por ejemplo, llegar a un destino concreto. Hablaremos en este caso de **agentes basados en objetivos**.

Agentes basados en objetivos y función de utilidad

La Figura 4 muestra la arquitectura general de un agente basado en objetivos.

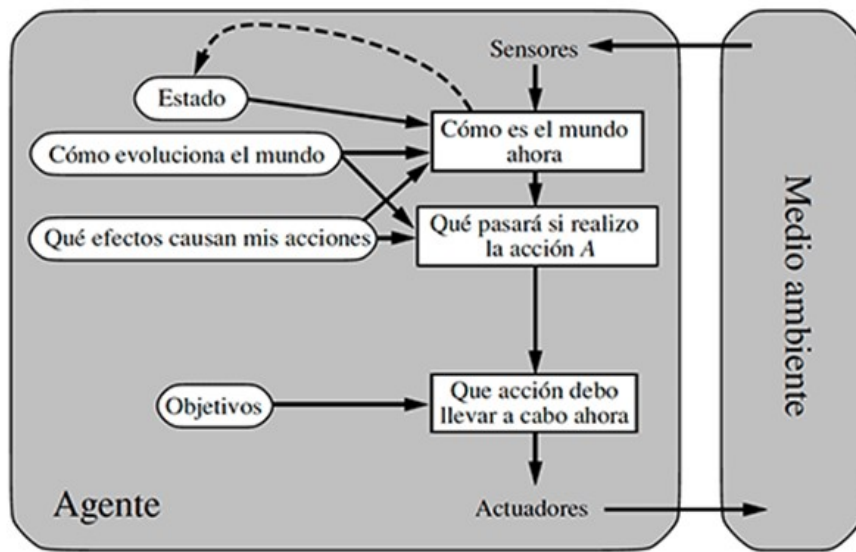


Figura 4. Agente basado en objetivos. Fuente: Russell, 2004, p. 57.

Este tipo de agentes son más flexibles, puesto que el objetivo que les guía obliga a representar explícitamente el conocimiento que soporta su decisión. Pero también son más complejos (dependiendo del objetivo planteado). Por ejemplo, el objetivo puede llegar con el coche a una determinada ubicación. Este tipo de tareas obliga a realizar una búsqueda de las distintas soluciones posibles y planificar un itinerario.

Los **agentes basados en objetivos** no permiten satisfacer nuestras necesidades en un gran número de ocasiones. Podemos pedir al navegador del GPS de nuestro vehículo que nos lleve a una ciudad concreta, pero nos sentiremos tremendamente defraudados y enfadados si nos damos cuenta de que nos está haciendo dar un rodeo innecesario o nos lleva por caminos poco transitables. Nos gustaría considerar no solo la meta a conseguir sino también en qué condiciones se debe alcanzar dicha meta para que consideremos plenamente satisfactoria la solución planteada por el algoritmo. Factores como llegar en el mínimo tiempo posible, recoger la mínima distancia posible, evitar peajes, etc., nos aporta valor sobre la utilidad de la solución.

Los **agentes basados en utilidad** incorporan una función de utilidad al modelo de tal

forma que es posible medir la satisfacción de los usuarios con la solución. Esta función de utilidad suele contemplarse como una función matemática a maximizar o minimizar. Por ejemplo, y como se ha comentado antes, se puede minimizar la distancia recorrida, minimizar el gasto en combustible, minimizar el tiempo invertido. Debemos notar que, matemáticamente hablando, un problema de maximización es similar a un problema de minimización, puesto que el $\max\{f(x)\}$ es igual al $\min\{-f(x)\}$.

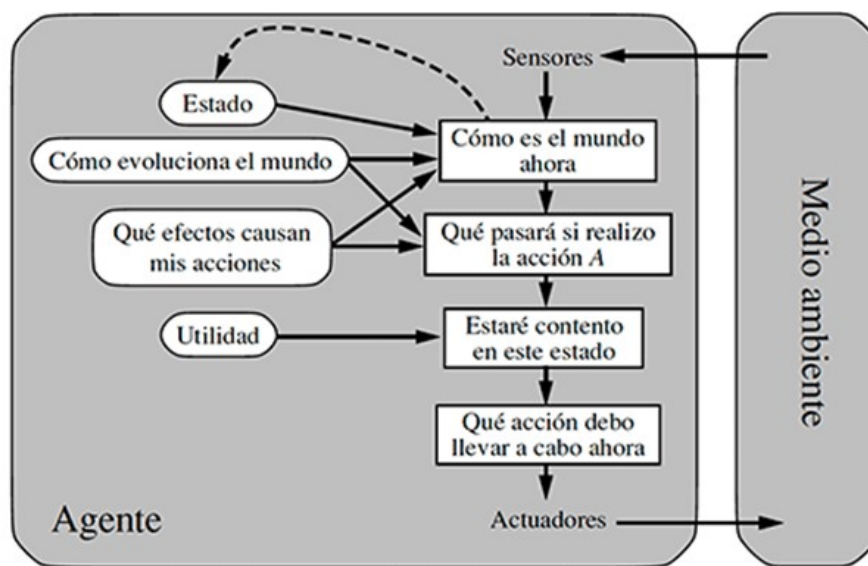


Figura 5. Agente basado en función de utilidad. Fuente: Russell, 2004, p. 59.

Se considera que un agente que posea una función de utilidad explícita puede tomar decisiones racionales con la ayuda de un algoritmo de propósito general que no dependa de la función específica de utilidad a maximizar o minimizar.

Con un **algoritmo de propósito general**, el tiempo de ejecución depende solamente del volumen o tamaño de los datos de entrada. Por el contrario, en un **algoritmo de propósito específico** el tiempo de ejecución depende de las propiedades asociadas a los parámetros de entrada y no tiene por qué estar directamente relacionado con el tamaño de los datos de entrada.

Agentes que aprenden

Los agentes explicados hasta ahora reciben «su conocimiento» del entorno a través de sus programadores. Son estos quienes introducen las reglas correspondientes, los objetivos a conseguir o la función de utilidad. Un enfoque diferente es enseñar a las máquinas a aprender por sí mismas las características de la tarea a abordar. De esta forma, la propia máquina podría ser capaz de descubrir nuevos enfoques y mecanismos eficientes para llevar a cabo la tarea asignada. Este tipo de agentes da pie a una nueva categoría, los **agentes que aprenden**.

Un agente que aprende se puede dividir en cuatro componentes conceptuales tal y como muestra la Figura 6 (Russell 2004).

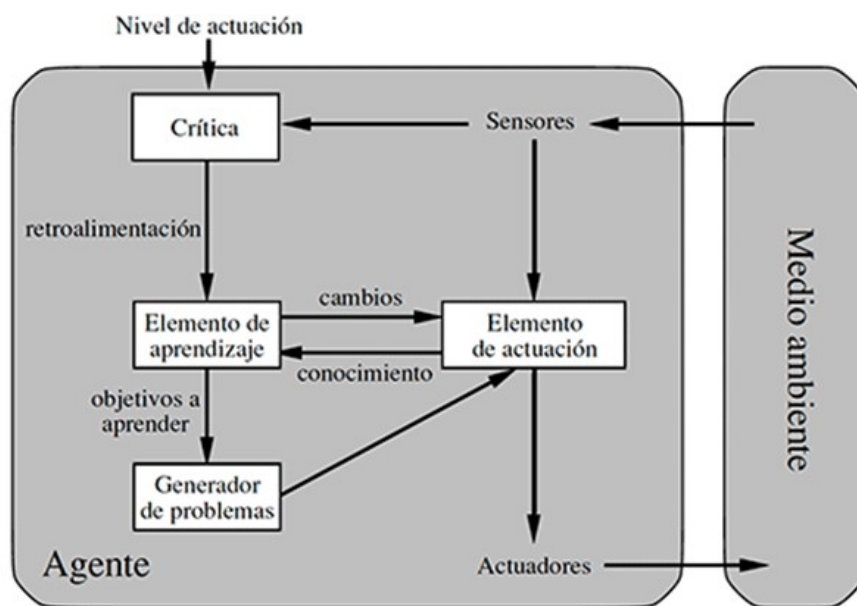


Figura 6. Estructura de un agente que aprende. Fuente: Russell, 2004, p. 60.

La actuación del agente genera críticas, entendidas estas como evaluaciones sobre la actividad del agente inteligente. Los propios sensores del agente se convierten en receptores de las críticas. Debemos entender estas críticas como una retroalimentación muy valiosa para evolucionar y mejorar el funcionamiento del agente.

La retroalimentación recibida alimenta el elemento de aprendizaje, que es el motor encargado de desarrollar mejoras. Los cambios realizados son compartidos con el elemento de actuación. Este último componente es el responsable de seleccionar las acciones a ejecutar considerando los estímulos recibidos a través de los sensores.

Por último, el **generador de problemas** tiene la misión de sugerir acciones exploratorias (similares a un experimento) que, aunque puedan ser consideradas como no óptimas a corto plazo, son útiles para llegar a soluciones mejores que las actuales a largo plazo.

La arquitectura reflejada en la última imagen debe entenderse como una arquitectura de alto nivel donde solo se refleja la composición e interacción entre los elementos principales del esquema. De hecho, como **elemento de actuación** debe entenderse una representación simplificada de lo que anteriormente se había considerado como el agente completo ya que recibe estímulos y determina las acciones a realizar. Una posible implementación de este tipo de agentes es usando **aprendizaje por refuerzo**.

Para acabar de entender el último tipo de agente explicado podemos acudir al manual *Inteligencia Artificial: un enfoque moderno* (Russell, 2004). Los autores nos plantean el caso de un **taxi automatizado**, sin conductor. Podemos suponer que el usuario entra en el taxi e introduce la ubicación de destino. Aquí, el elemento de actuación agruparía todos los conocimientos y procedimientos que tiene el taxi para seleccionar las acciones a ejecutar en base a los requerimientos del conductor. La parte crítica, gracias a la observación del entorno, proporciona al elemento de aprendizaje la información necesaria para aprender y evolucionar su comportamiento. Un ejemplo puede ser un cambio de carril que ha provocado que un conductor proporcione un toque de atención con el claxon. El análisis de dicha situación puede servir para que el vehículo

ajuste los parámetros de actuación ampliando el límite de separación con los vehículos circundantes antes de volver a realizar una operación de estas características. Este tipo de reacción no tiene por qué ser inmediata. Según el caso, se puede requerir un «volumen de críticas» previo determinado, antes de considerar un cambio de comportamiento.

El ciclo de las acciones está presente de forma constante en la filosofía de estos agentes. Mis acciones provocan reacciones en el mundo con el que interactúo y dichas reacciones provocan que ajuste mi comportamiento. Por supuesto, siempre teniendo en cuenta los principios básicos de diseño del agente, el objetivo a cubrir o función de utilidad.

Un ejemplo real de este esquema puede ser el motor de inteligencia artificial **AlphaGo** desarrollado por DeepMind (Google). En mayo de 2017, AlphaGo derrotaba a Ke Jie, campeón (humano) del mundo de *Go* por la mínima. Por primera vez en la historia, una máquina superaba a un campeón mundial en este milenar juego. AlphaGo sorprendió a su rival con movimientos imaginativos y poco usuales.

Las versiones iniciales de este tipo de juegos se basaban en el análisis histórico de millones de partidas previamente cargadas. AlphaGo basa su fortaleza en su capacidad de autoaprendizaje, siendo capaz de jugar millones de partidas contra sí mismo y aprender de todas las situaciones que se van produciendo durante el juego. De esta forma, no solo es capaz de descubrir por sí mismo diferentes trucos que los mejores profesionales usan de forma cotidiana en sus partidas, sino que es capaz de evolucionar las jugadas ideando acciones originales capaces de sorprender al rival.

8.5. Proyectos de investigación en agentes inteligentes

Los proyectos de investigación en agentes son un campo muy de moda y que tiene un enorme futuro por delante debido a que son aplicables a multitud de sistemas *hardware* y *software*.

Además, son fácilmente englobables en *topics* de investigación punteros como el *machine learning* ya que un agente puede estar implementado con modelos de sistemas expertos o aprendizaje automático. Y, de hecho, normalmente en los últimos años es lo que suele suceder. Sus principales campos de actuación más de moda son los siguientes:

- ▶ *Trading* automático en bolsa.
- ▶ Agentes de negociación automáticos (llegan a acuerdos sin transacciones, sin intervención de humanos).
- ▶ Internet de las cosas. Todo tipo de agentes que se conectan a internet y realizan operaciones de forma automática por ti.
- ▶ Drones autónomos.
- ▶ Vehículos autónomos.
- ▶ Robótica e industria 4.0.
- ▶ Videojuegos: comportamiento de NPC, jugadores de lucha entrenados por ti que juegan en red en tu nombre (*killer Instinct*, *Smash Bros. Ultimate...*).

En cuanto a publicaciones, las principales revistas sobre este tema son las siguientes:

- ▶ *IEEE Transactions on Automatic Control.*
- ▶ *Automática.*
- ▶ *IEEE Transactions on Systems, Man, and Cybernetics.*
- ▶ *Autonomous Agent and Multy Agents.*
- ▶ *System and Control lectures.*
- ▶ *International Journal of Control.*

8.6. ¿Qué son los sistemas expertos?

Un sistema experto es un **sistema que emplea conocimiento humano capturado por un analista especializado en la toma de requisitos de sistemas expertos**, que después se almacena en una computadora y que, mediante una serie de reglas y modelos, permite al sistema resolver problemas similares a los que puede resolver los expertos consultados. Algunos de estos sistemas imitan el proceso de razonamiento que los expertos utilizan para resolver problemas específicos, de esta forma estos razonamientos pueden ser validados por los propios expertos para comprobar que sean correctos. Normalmente **utilizan un sistema de razonamiento o bien inductivo o deductivo**. Aunque cada vez hay más sistemas que utilizan modelos basados en *machine learning* donde esta característica en ocasiones se ve más comprometida, debido a que muchos de estos algoritmos son de caja negra (opacos, no se conoce exactamente cómo han llegado a la solución que generan).

Existen multitud de tipos de sistemas expertos pero los principales los podemos agrupar en tres tipos.

Basados en reglas

Se captura información de los expertos y se modeliza en forma de reglas. Estas reglas tienen una estructura:

SI X Entonces Y

Donde X es el antecedente de la regla e Y el consecuente.

El antecedente para cumplirse debe tener en la base de hechos (la memoria del sistema experto) una instancia de un objeto que sea compatible con esta. Ser compatible significa que cumpla con la descripción planteada en el antecedente. Por ejemplo:

Si p: Paciente p.fiebre > 38 Entonces

p.enfermo = true;

Esta regla se activará para cada uno de los elementos p de tipo paciente que tengan el atributo fiebre mayor que 38 y los marcará como enfermos.

De esta forma encadenando reglas se va generando nuevo conocimiento:

Si p: paciente, p.enfermo && not p.ingresado && camaslibres > 0
entonces:

camaslibres--, p.ingresado=true

A esta forma de razonar se le denomina **encadenamiento hacia delante**. Y se basa en el proceso deductivo o en la aplicación de la regla *modus ponens*.

Este tipo de sistemas expertos suelen implementarse con sistemas de reglas de producción. Algunas herramientas para esto son Clips, Jess, Lisa o Prolog.

Basados en casos

Se le suele denominar *case base reasoning* y es un **sistema de razonamiento que se basa en la premisa de encontrar el caso más similar de una base de casos y ejecutar la acción que se realizó para ese caso en un momento anterior**. Estos sistemas han dado tradicionalmente muy buenos resultados debido a que son capaces de plantear soluciones a problemas con base en los criterios del experto. Pero tienen algunos problemas en ciertos entornos. En los métodos basados en razonamiento, el razonamiento realizado para llegar a la conclusión puede ser trazado y mostrado a un experto que valide este. Esto da mucha confianza en los resultados que produce el sistema. Sin embargo, en los sistemas basados en casos, necesitamos encontrar un caso que sea lo más parecido posible. Esto tiene como problema principal que la función de similitud de casos a veces no funciona correctamente y no identifica bien el caso más similar, lo que puede conllevar elegir

un caso que realmente no era tan parecido como pensábamos. Aquí razonar por qué un caso es más similar que otro de la base de casos no es tan claro como en el razonamiento deductivo, y puede haber problemas de trazabilidad de la toma de decisiones del sistema experto. Aun así, no llega a ser un algoritmo de caja negra ni mucho menos y es preferible cuando necesitamos una cierta explicabilidad a modelos más opacos como las redes de neuronas.

Se suele utilizar KNN (*k-nearest neighbors*) o modelos derivados de este para la selección del conjunto de casos candidatos de la base de datos a tratar. Donde K representa el número de casos que vamos a recuperar para su consideración.

Basados en redes bayesianas

Las redes bayesianas **representan un conjunto de variables aleatorias y sus dependencias condicionales a través de un grafo acíclico dirigido** (DAG por sus siglas en inglés). Por ejemplo, una red bayesiana puede representar las relaciones probabilísticas entre enfermedades y síntomas. Dados los síntomas, la red puede ser usada para computar la probabilidad de la presencia de varias enfermedades. Esto aporta algo que lo hace muy interesante y es el concepto de probabilidad. Al dar una probabilidad, el experto puede finalmente valorar entre diferentes opciones ordenadas según la más probable por el sistema experto y da también una idea del posible error que el sistema pueda tener. Cosa que es más complicada con los otros dos modelos.

Aunque hay versiones de sistemas expertos basados en reglas que utilizan lógica difusa. Este tipo de lógica sí que usa probabilidades y también puede aportar una idea de cuál es el error que puede tener el razonamiento obtenido.

Para ejecutar redes bayesianas existen diferentes sistemas *software*, por ejemplo, Winbugs, Jags u OpenBUGS.

8.7. Proyectos de investigación en sistemas expertos

Los proyectos de investigación en sistemas expertos se centran en las **siguientes ramas**:

- ▶ Medicina.
- ▶ Internet de las cosas.
- ▶ Cultivos.
- ▶ Ganadería.
- ▶ Economía.
- ▶ Visión por computador.
- ▶ Apoyo a las decisiones.

Estos proyectos se caracterizan por ser **entornos de aplicabilidad** donde el sistema normalmente apoya la decisión última de un humano y son sistemas que se basan principalmente en profesionales cualificados de los que extraen el conocimiento. Así que puede ser aplicado a cualquier entorno donde el conocimiento se pueda modelar a base de reglas, sistemas de inferencias o probabilidades. Actualmente los sistemas basados en **lógica difusa** son los que más activos están a nivel académico.

Las **principales publicaciones** de este medio son:

- ▶ *Expert Systems With Applications.*
- ▶ *International Journal of Information Management.*
- ▶ *IEE Access.*

- ▶ *Expert System.*
- ▶ *Decision Support System.*

8.8. Referencias bibliográficas

Russell, S. (2004). *Inteligencia artificial: un enfoque moderno*. México: Pearson Educación.

Mastering the game of Go with deep neural networks and tree search

Silver, D. et al. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587), 484-489. doi: 10.1038/nature16961. Recuperado de <https://storage.googleapis.com/deepmind-media/alphago/AlphaGoNaturePaper.pdf>

Artículo científico publicado en la prestigiosa revista *Nature* donde se exponen los fundamentos de AlphaGo.

Expert Systems

Andy. (17 de agosto de 2018). *What are expert system?* moral-robots.com.
Recuperado de <https://moral-robots.com/ai-society/what-are-expert-systems/>

Artículo interesante donde da una idea global de qué es un sistema experto.

1. ¿Cuáles de las siguientes frases son ciertas respecto a los agentes?
 - A. Solo pueden recibir información de forma directa de otros agentes.
 - B. Reciben información del entorno con el que se relacionan.
 - C. Sus acciones siempre afectan al entorno.
 - D. Su arquitectura se puede clasificar en diez tipologías distintas.

2. ¿Qué significa el acrónimo REAS?
 - A. Rendimiento, entorno, avisador, sensación.
 - B. Reutilización, entorno, acción, sensor.
 - C. Reutilización, entorno, acciones, sensores.
 - D. Rendimiento, entorno, actuadores, sensores.

3. ¿Cuáles de los siguientes pueden ser atributos de un entorno de trabajo?
 - A. Discreto.
 - B. Discrecional.
 - C. Multiepisódico.
 - D. Todas las anteriores.

4. ¿Cuáles de los siguientes pueden ser atributos de un entorno de trabajo?
 - A. Estocástico.
 - B. Parcialmente observable.
 - C. Episódico.
 - D. Todas las anteriores.

5. El entorno de trabajo en el que un agente inteligente debe desenvolverse para jugar al póker es un ejemplo de entorno de trabajo:
 - A. Parcialmente observable, secuencial y continuo.
 - B. Totalmente observable, secuencial y continuo.
 - C. Parcialmente observable, estático y discreto.
 - D. Parcialmente observable, dinámico y continuo.

6. Ejemplos válidos de distintos tipos de agentes inteligentes según su estructura son:
 - A. Reactivo simple, basado en modelos, basado en alcance.
 - B. Reactivo simple, reactivo simple basado en modelos.
 - C. Basado en objetivos, basado en función dinámica.
 - D. Agente que aprende, agente que juega.

7. Los agentes reactivos basados en modelos:
 - A. Tienen en cuenta el histórico de información para la toma de decisiones.
 - B. Solo tienen en cuenta un conjunto de reglas de condición-acción.
 - C. Consideran una función de utilidad.
 - D. Un agente de estas características ha derrotado al campeón mundial de *Go*.

8. Los sistemas expertos son útiles para:
 - A. Obtener conocimiento más allá del aportado por los expertos.
 - B. Suplantar al experto en la toma de decisiones.
 - C. Modelizar el comportamiento de expertos humanos para apoyar sus daciones.
 - D. Todas las anteriores.

9. El encadenamiento hacia delante:

- A. Es un método deductivo que se aplica en los sistemas expertos para obtener nuevo conocimiento a partir de una base de hechos y unas reglas de inferencia.
- B. Es el mecanismo por el cual funcionan los sistemas de producción.
- C. Permiten trazar el razonamiento del sistema experto.
- D. Todas las anteriores.

10. El razonamiento basado en casos:

- A. Es más fácil de trazar la respuesta del sistema que los algoritmos de encadenamiento hacia delante.
- B. Son un algoritmo de caja negra.
- C. Tienen peores resultados que los sistemas basados en redes bayesianas.
- D. Son más complejos de trazar que los sistemas de encadenamiento hacia delante.