Sesión 1 Tema IA y Sistemas Inteligentes

¿Qué es la inteligencia? Esta ha sido la primera cuestión planteada durante la primera sesión de la asignatura; Por la inteligencia entendemos factores como la memoria, el pensamiento, el lenguaje o la creatividad etc. Todos estos factores nos confirman que no existe un solo tipo de inteligencia, y por lo tanto no es solo una característica de los genios. Debido a todas estas trabas sobreentendemos que no podemos dar una definición clara de la inteligencia, sin embargo según Hofstadter la inteligencia sería la capacidad de responder flexiblemente, aprovechar las circunstancias, responder sentidos ambiguos, diferenciar y generar o crear conceptos o ideas nuevas.

Pasemos a la pregunta que nos atañe en la asignatura ¿Qué es la inteligencia artificial? Podríamos deducir que son todas aquellas máquinas que actúan como humanos, pero esto no es así, podríamos diferenciar la IA en dos tipos, IA fuerte un sistema inteligente que sea consciente de lo que procesa y que en sí existe, y las IA débiles, las cuales solo resuelven problemas complejos.

Alan Turing propuso realizar un test donde en dos salas separadas se introducían a un humano y a un computador, y si una humano se comunicaba con los dos y no podía distinguir quién es la máquina entonces se sabría que una máquina es inteligente. Ante este test se propuso su contraparte "La sala china" donde se le proporcionaría a un humano en una sala varios manuales de chino a ingles pero sin introducción y la prueba consistiría en llegar a traducir una frase en chino sin saber que es de ese idioma.

En cuanto a la historia de la IA, las bases de la IA moderna que tanto conocemos como los robots humanoides, los modelos evolutivos o incluso el Deep Learning, se asentaron en la década de los 50 hasta los 80, donde fue tan investigada como criticada. Podemos poner como ejemplo del "fenómeno" de la inteligencia artificial la aparición de la IA HAL en 2001 odisea en el espacio (1968) donde se veían sistemas que son ahora nuestro día a día.

Pero como hemos hablado en esta sesión,  tanto la inteligencia artificial como la tecnología seguirán creciendo día a día y no dejarán de sorprendernos como lo han hecho hasta ahora.

[Versión imprimible](https://moodle2021-22.ua.es/moodle/mod/wiki/prettyview.php?pageid=1827)

Sesión 1:INTELIGENCIA ARTIFICIAL 101 COSAS QUE DEBES SABER HOY SOBRE NUESTRO FUTURO

Como todos sabemos la inteligencia artificial ya esta siendo usada para ayudar a los humanos a beneficiarse de manera significativa y disfrutar de una mayor eficiencia en todos los ámbitos de la vida.

Algunas de las maneras de beneficiarse son con el reconocimiento de imágenes estáticas, clasificación y etiquetado, mejoras en el desempeño de la estrategia comercial, procesamientos predictivo, seguridad contra amenazas etc. Todas estas aplicaciones recientemente mencionadas vienen dadas por los distintos tipos de inteligencias descritas en clase y que todos los humanos poseen, por lo tanto podríamos decir que si una inteligencia artificial es capaz de dominar cada una de estas capas podríamos hablar de un computador verdaderamente inteligente.

Pero ¿Podrá la IA ver, oír y entender como los humanos? Para ello debemos saber que se han desarrollado tecnologías como la visión por computador que permite ver el entorno desde los ojos de un computador, el reconocimiento de voz que permitirá a las IA entender cada una de las palabras que se le dice y comprenderlas. Los científicos en inteligencia artificial están logrando nuevos avances sorprendentes en cada uno de estos tres campos. Por ejemplo, Google ha anunciado que la visión artificial que la empresa ha desarrollado ya puede añadir los colores apropiados a fotografías y vídeos que originalmente están en blanco y negro, lo que demuestra que una IA puede entender lo que esta viendo (un humano, un perro, una calle, un coche) y dotar todos los elementos de color.  
  
Enlace del artículo: <https://static0planetadelibroscom.cdnstatics.com/libros_contenido_extra/40/39308_Inteligencia_artificial.pdf>

Sesión 2: Estrategias de búsqueda

Para conocer que son las estrategias de búsqueda, primero deberemos conocer que problemas resuelven, en primer lugar podríamos hablar de la planificación de rutas, donde por ejemplo entrarían casos prácticos como la planificación de rutas aéreas, o la de redes de computadores, o más campos como la del diseño de proteínas.  
Definiremos estas estrategias de búsqueda como una transición de estados, donde pasaremos de un estado a otro comunicándonos entre los vecinos de nuestra posición actual, para ello utilizaremos la heurística, donde recorreremos un árbol de búsqueda eliminando repeticiones, hasta llegar a nuestro destino de la manera más optimas posible.  
  
Los sistemas de producción formaliza la búsqueda de datos, mediante Base de hechos, Reglas de producción y estrategias de control.  
Estos algoritmos comienzan en un estado inicial, y después repiten un bucle eligiendo el mejor de los estados, hasta que se llega a la condición  que termina el bucle.  
  
Pero como elegimos la EC, tendremos que hacer unos una serie de preguntas para ello deberemos hacernos una serie de preguntas, entre ellas "¿Se puede descomponer el problema?,¿Se puede deshacer una solución si tiene errores? ¿Se puede predecir?"  
  
Uno de los problemas más concurrentes es el siguiente:

El problema de las jarras de agua: Se tienen dos jarras de agua :Una de tres litros y otra de cuatro ,no disponen de marcas de medición, las jarras se pueden llenar y también vaciar.¿Cómo se puede lograr tener exactamente dos litros de agua en la jarra de cuatro?

Ahora describiremos algunas estrategias de control básicas:

→ E1 Exploración de la frontera.→ E2 Cálculo de reglas aplicables→ E3 Resolución de conflictos.→ E4 Aplicación de regla y memorización de estado.

E1->Elegimos el mejor de los estados. E2->Decidimos si las reglas son aplicables en el estado que hemos decidido.E3-> incorporar el conocimiento para decidir qué regla nos acercará más a la solución.E4->Aplicamos y actualizamos el árbol  
  
Dentro de estas estrategias dividiremos entre Tentativas que son aquellas donde las búsquedas pueden tener mono o multicamino es decir que se mantienen estados de vuelta atrás por si nos equivocamos E Irrevocables donde no se permite la vuelta atrás.

### Sesión 2: Búsqueda Heurística

Hasta ahora hemos hablado de los algoritmos de búsqueda desinformados que buscaban en el espacio de búsqueda todas las posibles soluciones del problema sin tener ningún conocimiento adicional sobre el espacio de búsqueda. Pero el algoritmo de búsqueda informado contiene una serie de conocimientos tales como qué tan lejos estamos del objetivo, costo de la ruta, cómo llegar al nodo objetivo, etc. Este conocimiento ayuda a los agentes a explorar menos en el espacio de búsqueda y encontrar de manera más eficiente el nodo objetivo.  
El algoritmo de búsqueda informado es más útil para espacios de búsqueda grandes. El algoritmo de búsqueda informado utiliza la idea de heurística, por lo que también se denomina búsqueda heurística.  
  
Función heurística: la heurística es una función que se utiliza en la búsqueda informada y encuentra la ruta más prometedora. Toma el estado actual del nodo como entrada y produce la estimación de qué tan cerca está el agente del objetivo. Sin embargo, es posible que el método heurístico no siempre dé la mejor solución, pero garantiza encontrar una buena solución en un tiempo razonable. La función heurística estima qué tan cerca está un estado de la meta. Está representado por h  y calcula el costo de una ruta óptima entre el par de estados. El valor de la función heurística es siempre positivo.  
  
La búsqueda heurística pura es la forma más simple de algoritmos de búsqueda heurística. Expande los nodos en función de su valor heurístico h . Mantiene dos listas, lista ABIERTA y CERRADA. En la lista CERRADO, coloca aquellos nodos que ya se han expandido y en la lista ABIERTA, coloca los nodos que aún no se han expandido.  
  
<https://www.javatpoint.com/ai-informed-search-algorithms>

### Sesión 3: A\* pseudo

Hay que recalcar que para trabajar de manera más cómoda con el algoritmo se trabajará desde una clase auxiliar a la que llamaremos Estado o Nodo, dentro de esta clase tendremos todas la componentes que atañen a nuestro algoritmo, estas componentes serán:  
Padre-> Esta componente nos dirá que nodo es el padre de que exploramos actualmente, cuando esta componente sea nula se considerará la raíz de nuestro árbol de decisiones.  
G-> Es el coste del camino elegido entre los nodos, es decir el coste que llevamos más el coste de llegar al nodo que queremos explorar  
H-> Es la cota optimista del algoritmo, se pueden realizar con diferentes heurísticas, entre ellas podríamos elegir la manhattan o podríamos actuar con desconocimiento poniendo su valor a 0  
f->Es el total de la suma entra la cota optimista y el coste de transición  
Una vez explicado todo esto podemos empezar con la explicación del algoritmo:  
Al principio del algoritmo tendremos dos listas una ListaInterior que estará inicialmente vacía y una listaFrontera donde añadiremos todos los nodos a los que podemos acceder desde el nodo origen.  
Ahora comenzaremos a explorar los nodos añadidos a la listaFrontera cogiendo el menor de todos los que tenemos disponible (compararemos los nodos mediante su coste), lo primero será comprobar si el nodo al que hemos accedido es el destino o no, si es el nodo destino deberemos terminar el bucle y mediante el acceso a sus padres y la suma de sus componentes G, pero si el componente que procesamos no es el destino deberemos pasar a examinar sus hijos, para ello eliminaremos el nodo de la listaFrontera y lo pasaremos a la listaInterior, ahora en los nodos que tiene alrededor buscaremos si tenemos alguno que este ya explorado, si no lo esta deberemos explorarlo y meterlo a la listaFrontera para su posterior exploración, si si que esta y el coste es menor deberemos actualizar su camino de menor coste

Sesión 5: Sistemas expertos

En la clase de hoy hemos hemos hablado de los sistemas expertos, los sistemas expertos son aquellos que mediante la inteligencia, el conocimiento y la inferencia, son capaces de entender, comprender y resolver problemas, reproduciendo los procesos de una mente humana.  
  
Sus comienzos fueron en los años 50 con el General Program Resolver y su auge fue en los 80 con el desarrollo de Clips.  
  
Estos sistemas se caracterizan por  su alto desempeño, tiempo de respuesta adecuado, confiabilidad, comprensible, flexibilidad, representación explicita del conocimiento, para ello utilizan reglas lógicas  de if then else, preposiciones y predicados, plantillas y  incertidumbre.  
  
Cuando entra la incertidumbre en juego se pierden las propiedades de modularidad y monotonía .  
  
Ahora hablaremos de los sistemas expertos difusos, en clase se nos ha puesto el ejemplo de lo que consideraría un humano ser joven y lo que consideraría una máquina, un humano consideraría joven un rango variable y difuso, ya que una persona  de 30 años es joven comparado con una de 50 y una persona de 21 años es igual o más joven que una de 21 años y 2 meses, pero para una máquina esto no sería así, ya que esta cogería un rango de entre pongamos 21 y 30 años y los 30 años y un mes dejaría de ser joven.  
  
Para paliar este tipo de desinformación aplicaremos la variables lingüísticas que son palabras o sentencias que van a enmarcarse en un lenguaje predeterminado, como por ejemplo tendríamos palabras como "Muy, más o menos, casi"

### Sesión 5: Juegos búsqueda

La búsqueda en juegos dio comienzo en 1760 con el "Ajedrecista mecánico" de Wolfgang Kempelen y fue evolucionando con Charles Babbage y su máquina analítica de ajedrez, esta vertiente de investigación en el campo del ajedrez hizo que tras varios intentos en 1997 la IA Deep Blue ganara al vigente campeón de ajedrez.

Pero los juegos generan problemas de búsqueda, ya que no se pueden generar todas las ramas del árbol de búsqueda, ya que hay un determinado nivel de profundidad, para ello se debía generar una función de evaluación, que genera un valor numérico de como de bueno es un estado, MAX maximiza la función y MIN la minimiza, en algunos casos el MAX nos devolverá si pierde, gana o empata. La raíz del árbol será el inicio de la jugada es decir MiniMax, el estado será la configuración en momento dado, Cada rama indica un posible movimiento, y cada rama completa una posible jugada.

Las estrategias de búsqueda MinMax, nos llevan desde generar todos los nodos hasta una determinada profundidad, evaluar cada nodo y asignar un valor nodo raíz., y asignar el valor Min al valor mínimo de sus hijos y el Max al máximo.

Por otro lado en los juegos multijugador, MinMax extendido a juegos de más de dos jugadores donde se sustituye el valor de un nodo por un vector de valores, tantos como jugadores,  Hoja es el estado desde el punto de vista de cada jugador.

Las estrategias de poda:

La poda alfa beta es una técnica de búsqueda que reduce el número de nodos evaluados en un árbol de juego por el Minimax.α es el valor de la mejor opción hasta el momento a lo largo del camino para MAX, esto implicará por lo tanto la elección del valor más alto β es el valor de la mejor opción hasta el momento a lo largo del camino para MIN, esto implicará por lo tanto la elección del valor más bajo.

Estrategias complementarias:

Movimientos de libro: Utilizada en aquellos momentos que mirando las siguientes ramas no se puede seleccionar un movimiento mirando la configuración actual de juego, para ello se enseña al programa una serie de movimiento básicos estudiados para mejorar el rendimiento del programa.

Espera del reposo: busca evitar el efecto horizonte, condición adicional de en recursión, se evalúa un nodo y este cambia su valor de forma drástica.

Bajada Progresiva: recorre los nodos por niveles, al llegar a la petición de la jugada devuelve la solución.

Poda heurística: reducir B desarrollando únicamente los mejores movimientos de cada nivel con bajo coste.

Continuación heurística: solo se puede tener conocimiento hasta la profundidad seleccionada, desarrolla la anchura hasta un determinado nivel, selecciona un subconjunto de nodos terminales para búsquedas más profundas, selecciona un conjunto de heurísticas seleccionadas con el juego