



Inteligencia Artificial (y II)

Más aplicaciones prácticas

David Arroyo Menéndez

En la primera parte de este artículo se introdujo las problemáticas que se abordaban en la inteligencia artificial. En la segunda parte se mostrará cómo conseguir que las máquinas hagan planes, cómo pueden comprender el lenguaje humano, cómo pueden pensar con neuronas y cómo aplicar todas estas cuestiones a tareas tales como la robótica, entre otras cosas, todo ello con un software libre.

Planificación

Llamaremos planificación al proceso de búsqueda y articulación de una secuencia de acciones que permitan alcanzar un objetivo. Por ejemplo, si nuestro objetivo es viajar desde un pueblo perdido de Asturias y queremos llegar a Guatemala la secuencia de acciones serían los distintos transportes que se deben tomar para llegar. Otro ejemplo podría ser que tuviéramos un robot en un laberinto y nuestro objetivo fuera sacarle de él; en tal caso, nuestras acciones serían los tramos recorridos en línea recta y los giros dados por el robot.

Para formalizar el problema de la planificación existen dos notaciones principales: ADL y STRIPS (http://www.rci.rutgers.edu/~cfs/472_html/Planning/STRIPS_plan_472.html). Ambas coinciden en utilizar la lógica para representar estado inicial, objetivo y acciones. También coinciden en que, para aplicar una acción, es necesario cumplir unas precondiciones y, tras haber ejecutado la acción, se habrá provocado unos efectos.

Para obtener la secuencia de acciones, es decir, el resultado al problema de la planificación, existen varios enfoques:

- Búsquedas en el espacio de estados que operan hacia adelante (desde el estado inicial), o hacia atrás (desde el objetivo) aplicando las acciones. También



Los robots pueden realizar tareas peligrosas. En la imagen, demolición controlada del edificio Windsor, en Madrid.

existen heurísticas (estrategias) eficaces que nos ayudan en la búsqueda. Estos enfoques funcionan bien cuando los subobjetivos son independientes.

- Si los subobjetivos no son independientes suele ser buena idea utilizar algoritmos de Planificación de Orden Parcial (POP), que exploran el espacio de planes sin comprometerse con una secuencia de acciones totalmente ordenada. Trabajan hacia atrás, desde el objetivo y añaden acciones para planificar cómo alcanzar cada subobjetivo.
- Otras estrategias prometedoras son el algoritmo GRAPHPLAN (<http://www-2.cs.cmu.edu/~avrim/graphplan.html>) y el algoritmo SATPLAN (<http://www.cs.washington.edu/homes/kautz/satplan/>).
- Cuando no se trabaja en mundos cerrados, sino en el mundo real, debemos tener en cuenta varias consideraciones. Muchas acciones consumen recursos: tiempo, dinero, materias primas, etc., por lo que debemos tener en cuenta medidas numéricas. Las redes jerárquicas de tareas (HTN)



(<http://www.cs.umd.edu/projects/plus/HTN/>) permiten tener sugerencias sobre el dominio por parte del diseñador. La información incompleta puede ser manejada mediante planificación que utilice acciones sensoriales para obtener la información que necesita. La planificación multiagente es necesaria cuando existen otros agentes en el entorno con los que cooperar, competir o coordinarse.

Los problemas y soluciones que se abordan en planificación tienen aplicaciones directas en gestión de tareas (workflow), control de misiones complejas (espaciales, satélites, militares, etc.), turismo (visitas a ciudades, planificar rutas...), procesos de enseñanza/aprendizaje, robótica (planificar caminos)...

Para la problemática de la planificación, las licencias libres han brillado por su ausencia. No obstante, todos los algoritmos se distribuyen con su código, lo cual también es cierto en arquitecturas integradas como PRODIGY (<http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/prodigy/Web/prodigy-home.html>), muy usadas en planificación.

Procesamiento de lenguaje natural

Antes de abordar qué es el procesamiento del lenguaje natural (o lenguaje humano), pensemos por un momento qué es la comprensión. Según Rich y Knight "comprender algo es transformarlo de una representación a otra, en donde la segunda representación se ha elegido para que se corresponda con un conjunto de acciones posibles que podrían llevarse a cabo, y en donde se ha diseñado la correspondencia de forma que para cada suceso se realice una acción apropiada". En definitiva, comprender algo es transformarlo en una representación que nos sea más útil para lograr nuestro objetivo.

¿Para qué podemos querer que una máquina comprenda lo que pone en un texto? Lo primero que se nos puede ocurrir es que sirva para comunicarnos con ella como si de una persona se tratara, pero ello no solo requeriría comprender nuestras palabras, sino también las emociones con las que las expresamos, lo cual es algo más complejo. La traducción automática desde un lenguaje natural a otro es algo para lo que solo necesitamos comprender el significado neutral de las palabras. Si conseguimos que un ordenador comprenda, también nos va a ayudar en tareas de recuperación de la información; me refiero a buscadores más eficaces que sepan relacionar mejor unas palabras o frases con otras, e incluso independientemente del idioma.

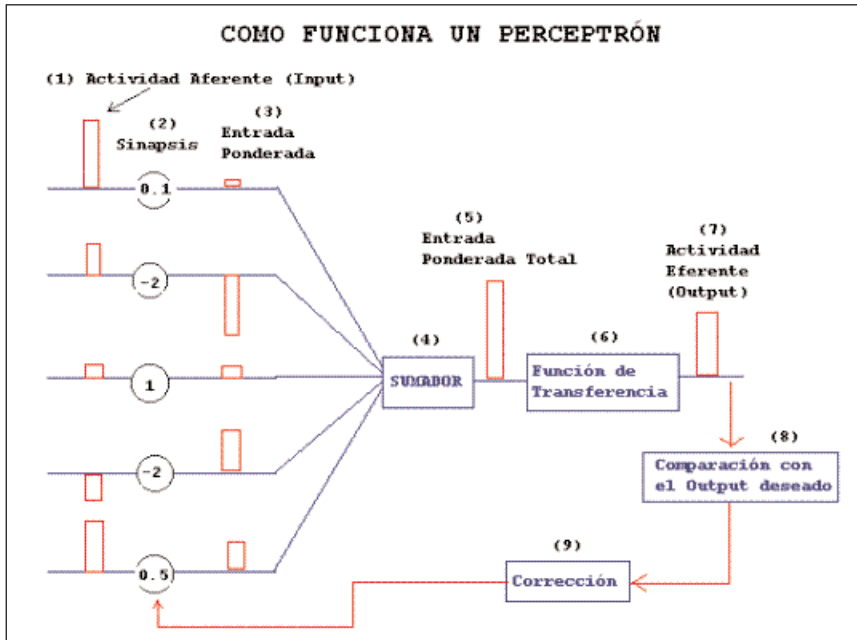
Para realizar estas tareas es necesario comprender un texto; esto es lo que nos proporciona el procesamiento del lenguaje natural. Veamos las fases de las que se compone [Rich y Knight, 1994]:

- **Análisis morfológico:** Se analizan los componentes de las palabras individuales y se separan de las palabras los constituyentes que no forman parte de ellas, como los símbolos de puntuación.
- **Análisis sintáctico:** Se transforman las secuencias lineales de palabras en ciertas estructuras que

muestran la forma en que las palabras se relacionan entre sí. Se pueden rechazar algunas secuencias de palabras si infringen las reglas del lenguaje sobre la forma en que las palabras pueden combinarse. Ej: "niña la come mucho" se rechazaría.

- **Análisis semántico:** Se asigna significado a las estructuras creadas por el analizador sintáctico. Es decir, se hace una correspondencia entre las estructuras sintácticas y los objetos del dominio de la tarea. Las estructuras en las que no se pueda realizar tal correspondencia se rechazan. Ej: "Las ideas verdes incoloras duermen furiosamente" se rechazaría.
- **Integración del discurso:** El significado de una frase individual puede depender de las frases precedentes y puede influenciar el significado de las frases posteriores. Por ejemplo, la palabra "lo" en "Jaime lo quiso" depende del contexto del discurso, mientras que la palabra "Jaime" puede influenciar el significado de frases posteriores como "Él vive en Madrid".
- **Análisis de la pragmática:** La estructura que representa qué se ha dicho se reinterpreta para determinar su significado actual. Ej: "¿Sabe qué hora es?" se reinterpreta como petición de hora.

nlTK en funcionamiento.



Esquema del funcionamiento de una neurona tipo perceptrón, obtenido de <http://www.geofisica.cl/>.

Hasta ahora hemos estado hablando del procesamiento del lenguaje escrito. Si quisiéramos realizar procesamiento del lenguaje oral, necesitaríamos conocimiento adicional sobre fonología, así como suficiente información adicional para manejar las posibles ambigüedades que pudieran surgir.

Existen varias herramientas libres para tratar el procesamiento del lenguaje natural. Podemos encontrar malaga-bin y mmorph. Sin embargo, creo que para jugar y/o iniciarse en este campo es más adecuado nltk (<http://nltk.sourceforge.net/>) (*Natural Language Toolkit*), escrito en python y con una licencia de creative commons (<http://creativecommons.org/licenses/by-nc-sa/1.0/>).

Computación neuronal

Bien, hasta aquí hemos visto que las máquinas pueden imitar comportamientos humanos para un gran número de tareas. De hecho, incluso pueden aplicarse algoritmos de aprendizaje que, para ciertas tareas como el spam, dan muy buenos resultados. La siguiente cuestión entonces es: ¿pueden las máquinas pensar tal y como lo hacemos los humanos?

La naturaleza nos ha dotado con un sistema de cómputo basado en una enorme red de neuronas. Si las máquinas pudieran simular tal sistema de cómputo ¿no estarían entonces en condiciones de pensar tal y como lo hacemos las personas?

En primer lugar, pensemos qué es una neurona. Una neurona está formada por el cuerpo celular y diferentes prolongaciones: el axón, por el que transitan los impulsos nerviosos o potenciales de acción desde el cuerpo celular hacia la siguiente célula, y la/s dendritas, con número y estructura variable según el tipo de neurona, y que transmiten los potenciales de acción desde las neuronas adyacentes hacia el cuerpo celular. Se podría decir que las dendritas son entradas y el axón es la salida.

Ahora bien, ¿cómo se decide si tras recibir entradas se produce o no una salida, y qué salida se produce? Pues bien, la salida es la suma ponderada de las entradas seguida de una función umbral. Esto puede parecer un diodo: si el potencial de las entradas alcanza un cierto valor, entonces se produce la salida, y sino no. Sin embargo, difiere en ser ponderado. Se refiere a que no todos los valores de las entradas van a tener la misma importancia de cara a decidir si se produce o no la salida.

Si nuestra neurona no es más que un diodo con entradas ponderadas, entonces juntando unos diodos con otros en forma de red ¿hará el tipo de procesos que realizamos los humanos como, por ejemplo, aprender? Es curioso que para aprender solo es necesaria una neurona, y el aprendizaje consiste en modificar el valor de los pesos (la importancia de cada entrada) aplicando un algoritmo a nuestro valor de salida. ¿Y cuál será ese algoritmo? Un psicólogo llamado Donald O. Hebb enunció una regla que dice que "las conexiones que unen diferentes neuronas que se encuentran activas en un instante dado se fortalecen". Se refiere a que si la neurona está transmitiendo, las entradas que están activas tendrán más importancia de la que ya tenían. Y esto es, más o menos, la esencia de la computación neuronal, especialmente la no supervisada (que no requiere de un tutor), y esta es la que nos encontramos en la naturaleza.

Pero en la práctica las redes neuronales artificiales no han creado en las máquinas comportamientos similares a los de las personas. Sin embargo, lo cierto es que los problemas que tradicionalmente son más complejos para las máquinas (visión artificial, reconocimiento del habla, etc.), se resuelven mejor utilizando redes neuronales que mediante otras perspectivas simbólicas.

La computación neuronal tiene como tarea genérica la clasificación. Por ejemplo, pensemos que queremos averiguar en qué imágenes está Wally y en qué imágenes no. Nuestras entradas serían las intensidades RGB de cada pixel de cada una de las imágenes, y tendría una única salida que nos diría si está o no.

Las redes neuronales tienen un modelo inherentemente distribuido, puesto que un procesador puede simular una neurona. Así mismo, son autoprogramables. La programación pasa por elegir un número de neuronas, las conexiones entre las mismas, un número de salidas, un número de entradas y una configuración inicial de pesos. Después bastaría con alimentarla correctamente y, si el aprendizaje es supervisado, darle la adecuada realimentación humana.

Como siempre, para profundizar recomiendo jugar con el software libre disponible. Lo que he encontrado ha sido *Fast Artificial Neural Network Library* (fann) (<http://fann.sourceforge.net/intro.html>), Genesis (<http://www.genesis-sim.org/GENESIS/>) y, aunque no totalmente libre, también creo importante destacar SNNS (<http://www-ra.informatik.uni-tuebingen.de/SNNS/>).

Finalizar diciendo que, si bien los algoritmos de redes neuronales no son otra cosa que algoritmos de apren-



dizaje automático, debido a las implicaciones filosóficas que tienen he decidido tratarlos aparte.

Robótica

[Russell y Norvig, 2004] Los robots son agentes físicos que realizan tareas mediante la manipulación física del mundo. Para ello se dotan de efectores, como pinzas, ruedas, brazos mecánicos, etc. También se equipan de sensores que les permiten percibir el entorno, como visores, sistemas de ultrasonidos, giroscopios, etc.

El interés de la IA en la robótica se centra en los agentes inteligentes que manipulan el mundo físico. No obstante, también es un campo de gran interés para aplicar otras ideas de IA como planificación o visión artificial (aplicar algoritmos de aprendizaje para el reconocimiento de formas).

Las áreas de aplicación de la robótica son múltiples: industria, agricultura, transporte, entornos peligrosos (donde no debería haber humanos), exploración (ej: viajes al espacio), salud (ej: cirugía gracias a la precisión milimétrica), entretenimiento...

Para experimentar en este área respetando nuestras libertades es posible encontrar programas divertidos para la simulación de comportamientos robóticos, como por ejemplo gnurobots (<http://directory.fsf.org/GNU/gnurobots.html>) o robocode (<http://robocode.alphaworks.ibm.com/home/home.html>). Este tipo de programas pueden llegar a dar mucho juego en la IA; de hecho, los bots del quake han sido materia de tesis doctoral (<http://www.kbs.twi.tudelft.nl/Publications/MSc/2001-VanWaveren-MSc.html>).

También es interesante salirnos un poco del software y meternos en algún proyecto de hardware abierto, como Microbótica (<http://www.microbotica.es/web/ha.htm>).

Agradecimientos y conclusiones

En primer lugar, agradecer a Ramiro Pareja, miembro de la rama de robótica del grupo agassi (<http://www.ieeesb.etsit.upm.es/ieee/index2.php?ojo=1>) sus orientaciones acerca de los usos de este área. Así mismo, agradecer a Antonio Rodríguez Anaya (<http://www.ia.uned.es/personal/arodriguez/>) sus comentarios acerca de planificación.

Referencias

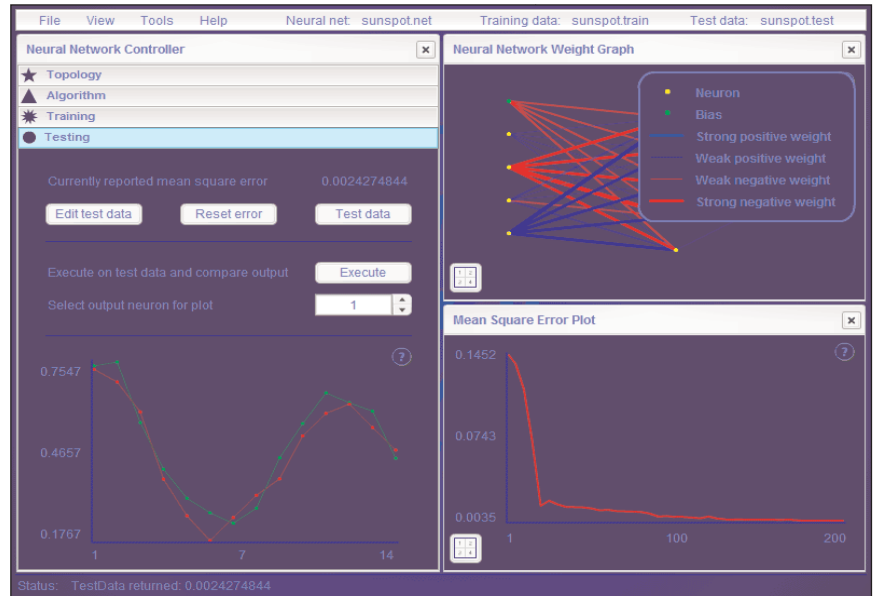
- Rich, E. y Knight K. (1994). *Inteligencia Artificial*. Madrid: McGraw Hill.
- Russell, S. J. y Norvig P. (2004). *Inteligencia Artificial. Un Enfoque Moderno*. Segunda ed. Madrid: Pearson Educación S.A.
- Howto de inteligencia y vida artificial: <http://linuxselfhelp.com/HOWTO/AI-AIife-HOWTO.html>
- Enciclopedia Libre: <http://enciclopedia.us.es>
- Página personal de Peter Norvig: <http://www.norvig.com>

Sobre el autor

David Arroyo Menéndez (<http://davidam.no-ip.info>) es Ingeniero Técnico en Informática de Sistemas por la UNED, donde estudia el segundo ciclo de dicha titulación. Actualmente trabaja en el Proyecto Europeo de Investigación Alfanet (<http://alfanet.ia.uned.es>).

Licencia

Derecho de Autor (c) 2005 David Arroyo Menéndez. Transcurridos cuatro meses tras su publicación este artículo podrá ser copiado y modificado libremente, siempre y cuando se haga explícito que el original fue publicado en *Mundo Linux* 77.



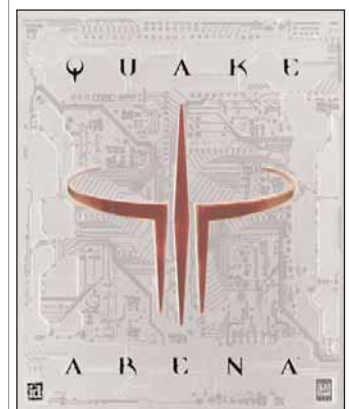
Un afectuoso agradecimiento a todo el equipo de *Mundo Linux* por la oportunidad de escribir este artículo y de poder hacerlo aplicando una licencia libre y muy especialmente a Margarita Padilla.

En el anterior artículo (ver *Mundo Linux* 75) se trataron aplicaciones más tradicionales o conocidas de la inteligencia artificial. En este segundo artículo, se desarrollan aplicaciones de mayor actividad investigadora. En ambos el software libre está presente con una fuerza notable dotando de herramientas al investigador o estudiante en múltiples áreas.

Si bien la inteligencia artificial aporta soluciones para casi todos las cuestiones que nos caracterizan como humanos e inteligentes, aún dista bastante para que realice dichas tareas de la misma manera y con la misma eficacia que lo hacen las personas. Sin embargo, las soluciones aportadas para resolverlos tienen aplicaciones interesantes en múltiples áreas de nuestra vida cotidiana que nos hacen la vida más fácil.

Espero que este artículo ayude a despertar el interés por la IA entre los usuarios de software libre y a que la comunidad que existe alrededor de la inteligencia artificial se interese aún más por el software libre. Al menos, sí espero que los lectores hayáis disfrutado tanto leyendo el presente artículo como yo escribiéndolo.

Fann en funcionamiento.



También se puede aprender de redes neuronales jugando al quake. Imagen obtenida de <http://www.amazon.com>.