

Computación Evolutiva: ¿Una Máquina de Movimiento Perpetuo para Información sobre Diseño?

Por Robert J. Marks II

La computación evolutiva, modelo derivado de la evolución darvinista, es una útil herramienta de ingeniería. Puede producir resultados ingeniosos, intuitivos e inesperados. Por consiguiente, la computación evolutiva se pinta frecuentemente como una fuente de inteligencia e información gratuitas. Sin embargo, para diseñar un programa que realice computación evolutiva se requiere infundirle información implícita acerca del objetivo del programa. Esta información afina la búsqueda evolutiva y la lleva a buen término.

Inteligencia Computacional

Hace cincuenta años, Ross W. Ashby preguntó: "¿Puede una máquina de ajedrez ganarle a su creador? (BRITISH JOURNAL FOR THE PHILOSOPHY OF SCIENCE [DIARIO BRITÁNICO PARA LA FILOSOFÍA DE LA CIENCIA], 1952). Hoy sabemos que sí puede. Una pregunta más relevante es: "¿Puede un programa de computadora generar más información que la que se le introduce?" Superficialmente, la computación evolutiva parece ser un candidato a paradigma; pero como sucede siempre que alguien asegura poder conseguir "algo a partir de nada", la respuesta es "no".

En la década de 1960, los pioneros de la computación evolutiva propusieron que la emulación computarizada de la evolución vencía la dificultad de demostrar la evolución darviniana en el laboratorio de biología. La comprobación de la evolución darviniana "se ha visto desfavorecida desde el principio por el hecho de que no se ha encontrado ningún experimento adecuado para decidir si tal evolución fue posible y cómo se desarrollaría bajo condiciones controladas". (N.A. BARRICELLI, ACTA BIOTHEORETICA, 1962). "En general, casi siempre es imposible o impráctico probar hipótesis sobre la evolución de una especie en particular mediante la preparación deliberada de experimentos controlados con organismos vivos de esa especie. Podemos intentar rodear parcialmente esta dificultad mediante la construcción de modelos que representen el sistema evolutivo que deseamos estudiar, y usarlos para probar por lo menos la validez teórica de nuestras ideas". (J.L. CROSBY, "COMPUTERS IN THE STUDY OF EVOLUTION" [LAS COMPUTADORAS EN EL ESTUDIO DE LA EVOLUCIÓN], SCI. PROG. OXF., 1967).

Diseño de Ingeniería

La computación evolutiva se utiliza hoy ampliamente en el diseño de ingeniería y la solución de problemas. El diseño empieza con el establecimiento de una meta u objetivo del diseño. A partir de una lista favorita de paradigmas, se elige un modelo viable. La actividad de diseño consiste en identificar valores de parámetros dentro del modelo elegido. El diseño se ha definido como "la manipulación sensata de valores de rango medio" dentro de los confines de un modelo (RANDALL JEAN, 2005). Los algoritmos de búsqueda hacen esto con la ayuda de una computadora.

Tomemos como ejemplo sencillo el diseño de una receta para hervir un huevo. Entre nuestras preguntas se encuentran las siguientes:

1. ¿Colocamos los huevos en agua fría y hacemos que hiervan o los colocamos en agua hirviente? (dos opciones)
2. ¿Por cuánto tiempo hervimos los huevos?
3. ¿Sacamos la cacerola del fuego y dejamos que el agua se enfríe? ¿Colocamos los huevos en un plato para que se enfríen? o ¿Metemos los huevos inmediatamente en agua fría? (tres opciones)

En el paso uno hay dos opciones; en el paso tres, tres. Con respecto a cuánto tiempo debemos hervir en el paso dos, vamos a asumir que hay opciones a intervalos de quince segundos desde 30 segundos hasta tres minutos: 0:30, 0:45, 1:00,, 3:00. Son once opciones. Entonces, el número total de recetas posibles es $2 \times 11 \times 3 = 66$. Hemos definido un *espacio de búsqueda*, pero aún no hemos definido nuestro criterio de diseño, a saber, ¿cuál es la receta óptima? Supongamos que pruebo el huevo y lo califico del uno al cien en cuanto al sabor. Esta medida, asignada a cada una de las 66 recetas, es la idoneidad de la receta. Cualquier calificación por encima de 90 cumplirá el criterio de diseño. La meta del diseño es la identificación de una receta que cumpla el criterio de diseño.

Asuma que nunca ha cocido un huevo y no tiene ni idea de cual receta es la mejor. Aplicamos *el principio de razón insuficiente de Bernoulli* que dice que en la ausencia de conocimiento previo, debemos asumir que todas las recetas tienen iguales probabilidades de ser la mejor. Debemos asumir que una receta es tan buena como las otras. Para encontrar la receta óptima, debemos probar todas las sesenta y seis. Un método para encontrar una receta decente es el de prueba y error. Si se pudiera hacer en computadora, podríamos terminar rápidamente. Suponga que podemos emular en computadora el hervor del huevo y la *idoneidad* de los resultados. Entonces podríamos determinar rápidamente la receta óptima mediante la evaluación de las sesenta y seis recetas. A la consideración de todas las soluciones posibles se le llama búsqueda exhaustiva. Desafortunadamente eso no es posible incluso con problemas de tamaño razonable, porque los problemas de búsqueda son difíciles de graduar. Si tuviéramos cien, en lugar de tres variables, y cada variable tuviera diez resultados posibles, el número de elementos contenidos por el espacio de búsqueda llegaría a ser 10^{100} (es decir, 10 multiplicado por sí mismo 100 veces), un número mayor que la cantidad de átomos existentes en el universo. En esos casos no es posible realizar una búsqueda exhaustiva.

La única forma de eliminar el principio de razón insuficiente de Bernoulli del problema de búsqueda es introducir información al proceso. La información puede ser explícita. En el caso de los huevos, los conocimientos de química nos dicen que colocarlos en agua fría inmediatamente después de hervirlos retarda la reacción química, lo que finalmente hará que huelan a azufre. Asumiendo que el olor a azufre va contra la idoneidad, podemos eliminar una de las variables de búsqueda y reducir las recetas a cuarenta y cuatro. Por otro lado, la información puede ser implícita. Por ejemplo, tal vez sepa que de diez cocineros famosos, dos colocan los huevos crudos en agua fría y ocho en agua hirviendo. Esta información puede guiar la búsqueda desde un principio hacia las recetas con mayor probabilidad de satisfacer el criterio de diseño.

Necesidad de Información Implícita

La teoría nos sugiere que, dada una computadora suficientemente rápida y el tiempo necesario, se puede buscar con éxito la solución óptima. Pero esto es como el mito de los "monos frente a la máquina de escribir". La historia, teóricamente creíble, dice que si una cantidad suficiente de monos bate letras al azar durante el tiempo suficiente, terminarán por producir todos los grandes textos de la historia. Es decir que si una cantidad suficiente de monos escriben letras al azar durante suficiente tiempo, obtendremos como resultado todos los grandes textos, como *Moby Dick* (1,170,200 caracteres), *Los Cuentos de los Hermanos Grim* (1,435,800 caracteres) y la *Biblia del Rey Jacobo* (3,556,480 caracteres sin incluir espacios). Sin embargo, el carácter finito del universo cerrado no permite esto.

A la búsqueda de una sola solución en un gran espacio de búsqueda no estructurado se le conoce como problema de búsqueda de una "aguja en un pajar". En casos moderadamente grandes, es sencillamente imposible. Eligiendo al azar letras del alfabeto inglés consistente de veintiséis letras, la probabilidad de llegar a escribir la Biblia del Rey Jacobo es de $26^{3,556,480} = 3.8 * 10^{5,032,323}$. Esta cifra es tan grande que nombrarla es un desafío. Si toda la materia del universo (10^{58} kg) fuera convertida en energía ($E = mc^2$) diez mil millones de veces por segundo desde el Big Bang (20 mil millones de años), y toda esta energía fuera utilizada para generar texto al mínimo nivel irreversible de bits (es decir, $\ln(2) kT = 2.9 * 10^{-21}$ joules por bit), entonces podrían generarse aproximadamente 10^{88} mensajes tan largos como la Biblia del Rey Jacobo. Si multiplicamos esa cifra por el número de átomos que forman el universo (10^{78}), obtenemos 10^{166} mensajes, una cantidad aún empedeñada por los $3.8 * 10^{5,032,323}$ requeridos.

Tratemos con un problema más modesto: la primera frase en inglés de la Biblia del Rey Jacobo

IN_THE_BEGINNING_GOD_CREATED (EN EL PRINCIPIO CREÓ DIOS)

(Podríamos completar la frase con "the heaven and the earth" (los cielos y la tierra), pero las cifras crecerían mucho). Hay 27 caracteres posibles (26 letras y un espacio) y una cadena de 28 caracteres. Las posibilidades de que los monos la escriban son de $27^{28} = 1.20 * 10^{40}$ a una. Esta cifra no es tan grande que no podamos abarcarla con nuestra mente. Las probabilidades de que un mono escriba 28 letras y *estas palabras específicas* son las mismas que las de elegir un sólo átomo entre más de un billón de toneladas cortas de hierro. [Usando el número de Avogadro, calculamos 27^{28} átomos (1 mole por cada $6.022 * 10^{23}$ átomos) (55.845 gramos por mole) (1 tonelada corta por cada 907,185 gramos) = $1.22 * 10^{12}$ toneladas cortas].

Las computadoras cuánticas ayudarían reduciendo el tamaño de búsqueda equivalente mediante una raíz cuadrada (HO et al., IEEE TRANS. AUT. CONT., Mayo 2003, p.783), pero el problema sigue estando más allá de los recursos del universo cerrado. Debe introducirse información al proceso de búsqueda.

Buscar en un espacio no estructurado sin imponer una estructura es computacionalmente imposible aún con problemas pequeños. Entre los primeros requerimientos de estructura se encontraban la disponibilidad gradiente, la dependencia de la solución óptima con respecto al segundo subproducto de la idoneidad, convexidad y funciones de idoneidad unimodal. (BREMNERMAN)

et al. "GLOBAL PROPERTIES OF EVOLUTION PROCESS" [PROPIEDADES GLOBALES DEL PROCESO EVOLUTIVO], 1966; NASH, "SUMT (REVISITED)", OPERATIONS RESEARCH, 1998). Recientemente, los teoremas llamados "no hay comida gratis" han recalcado la necesidad de información implícita impuesta por la heurística del diseño (WOLPERT, ET AL., IEEE TRANS. EVOLUTIONARY COMPUTATION [COMPUTACIÓN EVOLUTIVA], 1997). Estos teoremas han demostrado que "a menos que usted pueda hacer suposiciones preliminares acerca de los ... [problemas] en que esté trabajando, no puede esperar que ninguna estrategia de búsqueda, sin importar cuán sofisticada sea, pueda funcionar mejor que otra" (HO op. cit.). Los teoremas *no hay comida gratis* "indican la importancia de incorporar conocimientos específicos del problema al comportamiento del algoritmo [de optimización o búsqueda]" (WOLPERT, op. cit.).

Fuentes de Información

Una estructura común en la búsqueda evolutiva es la función de idoneidad impuesta, donde se le asigna un número al mérito de un diseño para cada grupo de parámetros. Entre mayor sea la idoneidad, mejor. El problema de optimización consiste en maximizar la función de idoneidad. *Las funciones de penalización* son similares, pero deben minimizarse. En los albores de la computación, un ingeniero colega mío describió su puesto en la conducción de búsquedas como *artista de la función de penalización*. Se sentía orgulloso de usar su área de erudición para crear funciones de penalización. El modelo de búsqueda estructurada desarrollado por el ingeniero en diseño debe ser, en cierto sentido, un *buen* modelo. La exploración de los parámetros de un mal modelo, sin importar cuán exhaustiva sea, no dará por resultado un diseño viable. Por el contrario, un modelo concebido ingeniosamente puede producir mejores soluciones en menos tiempo.

Este es un ejemplo sencillo de estructura en una búsqueda: en lugar de elegir cada letra al azar, seleccionemos con mayor frecuencia las letras más utilizadas. Si seleccionamos caracteres ingleses al azar, entonces la probabilidad de elegir cada carácter es de $1/27 = 3.7$ por ciento. En inglés, la letra "e" se utiliza aproximadamente el 10 por ciento de las veces. Los espacios ocurren el 20 por ciento de las veces. Si elegimos letras según su frecuencia de ocurrencia, entonces las probabilidades de elegir IN_THE_BEGINNING_GOD_CREATED caen en picada hasta *cinco millonésimas* (0.0005%) de su tamaño original -de $1.2 \cdot 10^{40}$ a $5.35 \cdot 10^{34}$. Sigue siendo un número muy grande: el billón de toneladas de hierro se ha reducido a 5 y media toneladas. Si utilizamos la frecuencia de los dígrafos, podemos reducirlo aún más. (Los dígrafos son pares de caracteres que ocurren con frecuencia; por ejemplo, el dígrafo "e_", donde "_" es un espacio, es el par de caracteres más común en inglés). La frecuencia de los trígrafos reducirá todavía más las probabilidades.

La Afinación del Espacio de Búsqueda

Conforme más implícita sea la estructura impuesta al espacio de búsqueda, más fácil será ésta. Lo más interesante es que, en el caso de mensajes moderadamente largos, si el mensaje meta no concuerda con la estructura del espacio de búsqueda, será imposible encontrarlo. (PAPOULIS, PROBABILITY, RANDOM VARIABLES AND STOCHASTIC PROCESSES [PROBABILIDAD, VARIABLES ALEATORIAS Y PROCESOS ESTOCÁSTICOS], 1991).

Teorema de la afinación del espacio de búsqueda. Supongamos que un espacio de búsqueda está estructurado de forma que genere un tipo de mensaje. Si un objetivo no concuerda con esta predisposición, las probabilidades de encontrarlo serán de cero.

Este teorema, conocido desde hace mucho en teoría de la información en otro contexto, es una consecuencia directa de la *ley de los grandes números*. Por ejemplo, si estructuramos el espacio de búsqueda para que produzca una "e" 10 por ciento del tiempo, entonces el número de "e's" en un mensaje de longitud 10,000 será muy cercano a 1000. Las probabilidades de encontrar el poco común libro *Gadsby*, que no contiene "e's", serían casi de cero.

La estructuración del espacio de búsqueda también reduce su tamaño efectivo. El espacio de búsqueda consiste en todas las secuencias posibles. En el caso de un espacio estructurado, llamemos subconjunto al conjunto de todas las secuencias probables predisuestas a la estructura del espacio de búsqueda. Para una estructuración dictada por la frecuencia de ocurrencia del alfabeto, todas las grandes novelas que busquemos, excepto la de *Gadsby*, caen dentro o cerca de este subconjunto.

Entre más estructura se agregue al espacio de búsqueda, se agrega también más información. Los trígrafos, por ejemplo, agregan más información que los dígrafos.

Teorema del subconjunto que se reduce. Conforme aumenten la longitud de una secuencia y la información de estructuración agregada, el porcentaje de elementos del subconjunto de búsqueda tiende a cero.

Por lo tanto, la estructuración de un espacio de búsqueda no sólo confina las soluciones a la estructura del espacio; conforme aumenta la longitud del mensaje, el número de soluciones se convierte en un porcentaje cada vez menor del espacio de búsqueda. (IBID).

Consideraciones Finales

Es necesario estructurar los espacios de búsqueda para que los algoritmos de búsqueda sean viables. Esto se aplica a la búsqueda evolutiva de una meta de diseño. También es necesario infundir implícitamente al espacio de búsqueda la información de estructuración que guíe el proceso hacia un resultado deseado. La meta puede ser específica, como una frase identificada con precisión, o general, como las frases que van a pasar, digamos, una revisión de ortografía y gramática. En cualquier caso, aún no existe ninguna máquina de movimiento perpetuo que diseñe la información surgida de la computación evolutiva.

RESUMEN BIOGRÁFICO: Robert J. Marks II es Profesor Distinguido de Ingeniería y Director del Departamento de Ingeniería en el área de maestría de la Universidad de Baylor. Es miembro de IEEE y la Sociedad Óptica de Norteamérica. El profesor Marks ha recibido la Medalla Centenaria de IEEE. Ha servido como Conferencista Distinguido para la Sociedad de Redes Neuronales de IEEE y la Sociedad de Inteligencia Computarizada de IEEE. El Dr. Marks fue el primer Presidente del Consejo de Redes Neuronales de IEEE (hoy sociedad). Tiene más de 300 publicaciones, algunas de ellas son muy buenas. Ocho de los documentos del Dr. Marks han sido reproducidos en colecciones de documentos destacados. Tiene tres patentes norteamericanas en el campo de las redes neuronales artificiales y procesamiento de señales.