

Apología de los algoritmos genéticos

Rafael Peñaloza Nyssen

Imagina que quieres optimizar (maximizar o minimizar) una función, de la que tienes poca o nula información, sobre un dominio finito. La restricción del dominio finito hace que los métodos de cálculo (léase derivación) no puedan ser aplicados; entonces, ¿cómo resolver el problema?

Una aproximación muy utilizada es el método probabilista. Si se generan aleatoriamente, por ejemplo, 100 elementos en el dominio (donde cada elemento puede ser generado con la misma probabilidad e incluso permitiendo repetición si se quiere) y se toma a aquel elemento dentro de los 100 generados que optimice la función sobre la muestra, entonces su valor funcional tiene cierta probabilidad (mayor a cero) de ser el óptimo sobre todo el dominio. Pero no hay necesidad de generar únicamente 100 elementos, se pueden generar 1000 o 1000000 o los que se quieran. La probabilidad de que aquel elemento que optimice la función sobre la muestra sea el óptimo global se acerca más y más a 1 a medida que el número de elementos generados es mayor. Esto se conoce como convergencia en probabilidad.

Por otro lado, no tienen por qué generarse todos los elementos de la muestra a la vez, o en un solo paso. Otra forma de hacerlo es generar una muestra pequeña, por ejemplo de 50 elementos, tomar el elemento que optimice a la función y luego generar otra muestra, agregarle el elemento que se tomó de la muestra anterior y elegir al que optimiza a esta nueva muestra (en este caso, esto es equivalente a haber tomado una muestra inicial de 100 elementos y elegido al óptimo sobre ella). El mejor individuo de esta nueva muestra es al menos tan bueno como el de la muestra anterior, pues este fue agregado a la muestra. Luego se repite el proceso para generar tantos elementos como se quiera.

Esta nueva perspectiva permite detener el proceso antes de alcanzar el tamaño de muestra especificado, si se alcanza un criterio de paro. Un ejemplo de tal criterio puede ser que el mejor valor de la función objetivo no cambie de un paso al siguiente.

Llamemos ahora "individuos" a los elementos de la muestra y "generación" a cada vez que se repite el proceso de generar el conjunto de individuos y escoger al mejor. Entonces, como ya se dijo antes, el óptimo de la última generación converge en probabilidad al óptimo global cuando el número de generaciones crece.

Otra mejora que se puede hacer al método consiste en cambiar la búsqueda ciega (generar uniformemente elementos del dominio en cada generación) por una búsqueda más dirigida. Esto se logra aumentando la probabilidad de generar un elemento del dominio si éste se encuentra "cerca" (bajo algún criterio de distancia) del mejor individuo de la última generación. De esta forma se exploran más a fondo las zonas "prometedoras". Cuando se hace esto, hay que tener cuidado de no volver muy específica la búsqueda, sino que exista la posibilidad de generar elementos lejanos al mejor elemento actual. Esto permite salir de valles que no llevan al óptimo en funciones cuya estructura es muy compleja.

Todo el proceso que se acaba de describir es conocido actualmente como "algoritmo genético" (o en forma más general, "algoritmo evolutivo"). Un algoritmo genético parte de una primera generación de individuos, para los que calcula el valor funcional (llamado "aptitud" del individuo). Después se produce aleatoriamente una segunda generación, creando artificialmente al mejor individuo (el más apto) de la generación anterior; proceso conocido como elitismo. De hecho no es necesario introducir de forma artificial únicamente

al mejor, sino que se pueden generar los 10 mejores, o cuantos se quiera; sin embargo, para asegurar la convergencia en probabilidad, es necesario que se mantenga al menos a uno. Si no se hiciera, el método consistiría únicamente en generar números aleatorios una y otra vez, olvidando lo que se hizo en la generación anterior (búsqueda sin memoria); esto provoca que el mejor individuo de una generación pueda ser peor que el mejor de la generación anterior.

Para la búsqueda dirigida se utiliza un concepto de distancia que tiene que ver con la diferencia en la forma de representar a los elementos del dominio. Por ejemplo, si se tiene una representación binaria y el dominio son los números enteros entre 0 y 16, entonces el número 8 (representado como 1000) es muy cercano a 12 (1100) pero lejano a 7 (0111). Esto se debe a que los elementos de un dominio no tienen que ser números, sino que pueden representar alguna otra cosa. La búsqueda dirigida se realiza utilizando los llamados operadores de mutación y cruce y los elementos de la generación anterior. Existe una gran diversidad de este tipo de operadores, pero todos se basan en el mismo principio, por lo que sólo describiré el más simple de cada uno.

El operador de cruce escoge dos elementos de la última generación (generalmente la elección se realiza de acuerdo a su valor funcional) y, con una cierta probabilidad definida de antemano, "cruza" estos elementos o los copia idénticos a la nueva generación. Cuando los "cruza", elige una posición aleatoria de la representación binaria y los datos a partir de esa posición entre los elementos; los nuevos elementos se colocan en la nueva generación. Este operador se aplica tantas veces como sea necesario para tener la cantidad de elementos que se requiere en la nueva generación.

El operador de mutación toma un factor entre 0 y 1 que representa la probabilidad de que cada 0 o 1 en la representación binaria se cambie por un 1 o 0, respectivamente. Por ejemplo, si este factor es 0.01, entonces cambiarán aproximadamente 1 de cada 100 dígitos de la representación binaria. Este operador se aplica sobre los elementos que se obtuvieron tras aplicar el operador de cruce.

Dependiendo de la representación utilizada (y de lo que ésta represente), es posible que los individuos generados utilizando estos operadores no sean válidos; es decir, que no se encuentren en el espacio de búsqueda. En esos casos se debe implementar un proceso de validación. Después, se pueden sustituir todos esos individuos por otros generados aleatoriamente, o modificarlos para que sean parte del espacio de búsqueda.

Entonces, un algoritmo genético no es más que una búsqueda probabilista dirigida. La aportación principal que dieron las ciencias de la computación en esta área no consiste en haberla relacionado con términos biológicos como "gen" o "evolución", ni el haberla relacionado con la teoría de Darwin y la supervivencia del más "apto"; sino en haber hecho notar que el método no se debe aplicar necesariamente a números. Con el avance de la computación y la representación binaria de todo tipo de elementos, se descubrió que el mismo método era aplicable a cualquier cosa representada de esta forma; no importa si se está hablando de números, pasteles, rutas óptimas o compilación de programas, lo único que importa es que cualquier elemento del espacio de búsqueda se pueda representar por medio de un "gen" al que se pueda aplicar el método.

Obviamente, la representación binaria no es la única posible. Por ejemplo, si se busca la expresión matemática que mejor aproxime a una secuencia finita de números como función de su posición, entonces la representación puede utilizar los símbolos $x, 1, 2, \dots, 9, (,), +, *, -, /$ o se pueden agregar más si se quiere. En este caso, sólo hay que adaptar los operadores de

mutación y cruce e incorporar un proceso de validación que elimine las expresiones que no tengan una sintaxis adecuada.

El apoyo de las ciencias de la computación también se aplica a la optimización de funciones definidas sobre un dominio continuo, pues no hay que olvidar que, a pesar de la enorme precisión que se pueda obtener dentro de una máquina, un número se representa siempre por medio de una cadena finita de símbolos y, por lo tanto, se tiene simplemente un dominio finito, aunque enorme, sobre el que se optimiza la función numéricamente.

Desafortunadamente, los científicos de la computación han perdido el camino. Cegados ante la idea de que habían desarrollado algo completamente novedoso, ante cualquier minúsculo cambio en el método surgía un “nuevo método” (¡y no se atreva alguien a confundirlos!) aún cuando en el fondo ninguno de ellos es más que una búsqueda probabilista. Por ejemplo, cada si se utiliza un operador de cruce distinto, el nombre del método cambia de “simple” a “de Vasconcelos” o “ecléctico”. Después se concibió el nombre "Algoritmo Evolutivo" para englobar todos esos métodos; sin embargo, ante el veloz desarrollo de la computación, este nombre va a ser también insuficiente pronto.

Por otro lado, tampoco hay que dejarse llevar por la impresión de que cualquier problema, en cualquier circunstancia, puede ser resuelto por medio de un algoritmo evolutivo. El método converge en probabilidad al óptimo dentro del espacio de búsqueda, sin embargo ese espacio de búsqueda depende de la representación (una mala representación puede dejar fuera a todos los elementos cercanos al óptimo global). Por su parte, el número de generaciones que se requieren para poder tener una certeza bastante alta de estar en el óptimo puede ser demasiado elevado para permitir al algoritmo operar durante el tiempo necesario para generarlas a todas. De hecho, para obtener el mayor potencial de un algoritmo genético, hay que definir todos los parámetros de acuerdo al problema en cuestión; y para esos mismos parámetros siempre es posible construir un ejemplo cuya solución sea inalcanzable en la práctica. Así, no hay UN algoritmo genético capaz de resolver cualquier problema.

Los algoritmos evolutivos son herramientas muy útiles, y abren la posibilidad de resolución de problemas que podríamos no saber cómo enfrentar. Sin embargo no hay que dejarse llevar por nombres y justificaciones artificiales, sino que hay que entender realmente lo que se encuentra detrás del método; esto permite, además, utilizar estos métodos de forma más clara y eficiente.