Cambios de paradigma de computabilidad para la resolución de problemas actualmente no computables: Estado del arte de la hipercomputación

Rolando J. Andrade Fernández
White paper
Caracas, Venezuela
contact@rolandoandrade.me

***Resumen*—La hipercomputación permite computar funciones que una máquina de Turing convencional no puede computar. La máquina de Turing se ve limitada al estar atada a la definición de algoritmo, y la computabilidad a su vez, por estar atada a la máquina de Turing, por lo que surgen movimientos que proponen una teoría relativista de la computabilidad como la hipercomputación, de la cual al principio salieron muchos modelos filosóficos-lógico-matemáticos como crítica a la teoría de computabilidad que existía, pero que gracias a los nuevos descubrimientos ha empezado a convertirse en algo muy real.**

***Palabras clave*—teoría de la computabilidad, hipercomputación, computabilidad relativa, problemas no computables, computación cuántica, inteligencia artificial, estado del arte**

I. Introducción

Alan Turing en su trabajo como matemático y uno de los fundadores de la computabilidad, se encargó afrontar muchos problemas que surgían a la hora de intentar automatizar la resolución de problemas de decisión por medio de máquinas, así él establece, que los problemas son en realidad funciones, y que algunos pueden ser computables y otros no, surgiendo el término hipercomputación como la capacidad computar funciones que una máquina de Turing no es capaz de computar [1].

De los problemas de computabilidad surgieron muchos retos, pero a su vez, se han propuesto bastantes soluciones, originándose tanto grupos a favor, como en contra de la posibilidad de la hipercomputación, por lo que se dará un breve contexto y antecedentes sobre el tema.

Una vez se tenga el contexto del problema, se va a continuar con el estado del arte de la hipercomputación, intentando explicar los modelos y avances que han existido en el área, los problemas que quedan por afrontar y los fundamentos para retractar los logros de la misma.

Finalmente ser realizará una valoración objetiva sobre el tema y la repercusión que éste podría tener en el futuro inmediato.

II. Antecedentes

II-A. *Funciones y algoritmos*

Alan Turing era un matemático y montó toda la teoría de la computabilidad sobre las matemáticas, atacando los problemas como funciones, las cuales se pueden resumir como "Se dice que y es una función de x cuando a cada valor de la variable x corresponde un valor único de la variable y" [2]. De esta manera, se puede ver una función como una caja negra donde a una entrada se obtiene una salida, tal como lo hace una computadora. Se puede ver una representación de ésto en la Figura 1.

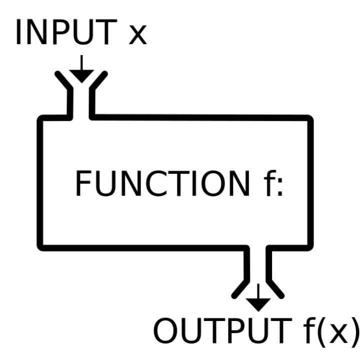

Figura 1. Ejemplo de una función como una máquina de [3]

A su vez, para que un computador lleve a cabo una tarea o la resolución de un problema específico es necesario indicarle qué operaciones realizar, en otras palabras, se tiene que describir cómo debe resolver el problema. Esa serie de pasos lógicos se denomina algoritmo [4] y a nivel computacional vienen representados como funciones, donde las computadoras reciben entradas y se obtienen salidas, mediante un proceso de actividades efectivas, de la misma forma en como representaba los problemas Alan Turing [5].

II-B. *La teoría de computabilidad y computación*

A finales de 1936 y principios de 1937, se publicaron dos artículos independientes de manera casi simultánea que dieron origen a la computabilidad, uno publicado por Alonzo Church[1] y otro por Alan Turing[2], donde los conceptos expuestos introducen al término *computabilidad* para su uso en procesos de actividades efectivas o sistemáticas o mecánicas [5].

Efectivas, sistemáticas o mecánicas se refieren a lo mismo, variando solo en la disciplina donde se use, lo que importa, es que para que una actividad M sea alguna de éstas debe cumplir con los siguientes requisitos [6].

- "M se establece en términos de un número finito de instrucciones exactas", es decir, se pueden escribir los pasos para llegar a un resultado.
- "M, si se lleva a cabo sin error, producirá el resultado deseado en un número finito de pasos", por lo que se puede concluir que a una entrada un resultado.
- "M puede (en la práctica o en principio) ser realizado por un ser humano sin ayuda de ninguna maquinaria, excepto papel y lápiz"
- "M no exige perspicacia, intuición o ingenio por parte del ser humano que lleva a cabo el método". El resultado puede hallarse sistemáticamente.

Bajo esta definición, surge la tesis más controvertida en el tema de hipercomputación, la tesis Church-Turing. La tesis de Turing es que "Las LCM[3] pueden hacer cualquier cosa que pueda describirse como 'regla general' o 'puramente mecánica'" [6]. Pero para entenderlo hace falta saber, qué hace específicamente una máquina de Turing.

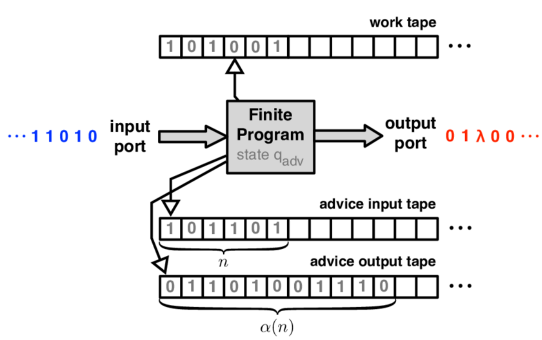

Figura 2. Ejemplo de una función como una máquina [7]

Una máquina de Turing es un modelo teórico, que puede considerarse una cinta infinita dividida en casillas, cada una de las cuales contiene un símbolo. Sobre dicha cinta, actúa un dispositivo que puede adoptar diversos estados y que, en cada instante, lee un símbolo de la casilla sobre la que está actuando. En función del símbolo que ha leído y el estado en el que se encuentra, podrá pasar a un nuevo estado, escribe un símbolo o se detiene [8]. En la Figura 2, se puede ver trasladado el concepto a las computadoras actuales, las cuales funcionan bajo la misma terоría que una máquina de Turing.

La teoría de la computabilidad es muy rígida, y asocia directamente el concepto de computabilidad con las máquinas de Turing [6], con lo que se establece que los problemas si no pueden ser resueltos por una máquina de Turing, no se pueden resolver con un algoritmo y por lo tando pasa a ser un problema indecidible a partir de la lógica matemática, por lo que son irrealizables para una computadora [5]. El mismo concepto de algoritmo limita en cierta forma, la esencia de la computación, estando subyugada a actividades efectivas y resultados absolutos.

II-C. *La corriente relativista y el surgimiento de la hipercomputación*

Los problemas no computables a lo largo de la historia han dejado a más de uno descontento, por lo que han surgido varias corrientes que tratan de complementar, desmontar o renovar la teoría actual de computabilidad. Entre las teorías que están ganando más fuerza, está la teoría relativista computacional, lideradas por el movimiento de hipercomputación que acuñó Copeland a principios del milenio [8].

Pese a que el nombre es relativamente nuevo, tan solo dos años después de la publicación de Turing ya se estaba empezando a hablar sobre una vía alterna al mundo de la computabilidad, pues Turing ligó el terminó computabilidad con las máquinas de Turing, haciendo el concepto de computabilidad absoluto, por lo que, o un problema es computable por una máquina de Turing, o sencillamente no es computable [9]. El mismo Copeland dice:

> "La tesis M[4] admite dos interpretaciones, de acuerdo a lo que sea que signifique 'puede ser calculado por una máquina', pudiendo tomarse bajo el concepto de 'puede ser calculada por una máquina que esté conforme a las leyes físicas (sin limitaciones de recursos) del mundo actual', o tomarse en un sentido ampliado, donde el problema se abstrae de la cuestión en que si la máquina existe o no, la hipótesis de una máquina que podría existir en el mundo actual. La versión estrecha de la tesis M es una proposición empírica cuyo valor de verdad es desconocido. La versión extensa de la tesis M es conocida para ser falsa. Varias máquinas hipotéticas que han sido descritas pueden calcular problemas que no son computables por una máquina de Turing"

La corriente relativista busca extender el concepto de computabilidad, pues no puede aceptar la idea de que existan problemas a los que no se le pueda extraer una solución, por lo que cree firmemente, que en realidad la computabilidad por máquinas de Turing en realidad es un conjunto, pero que los problemas no computables por máquinas de Turing,

[1]An Unsolvable Problem of Elementary Number Theory, The American Journal of Mathematics, vol. 58, pp. 345-363.

[2]On Computable Numbers, with an Application to the Enstscheidungsproblem, Proceedings of the London Mathematical Society, ser. 2, vol. 42, pp. 230-265.

[3]Máquinas de computación lógica: la expresión de Turing para máquinas de Turing

[4]La tesis de Church-Turing que se mencionó anteriormente

en realidad son computables relativos a otros conjuntos de computabilidad por que apliquen mecanismos distintos [10].

Este movimiento entonces define a los problemas no computables, como *no Turing computable* y está basada en arquitecturas de computadores hipotéticas que podrían resolver los problemas no computables y cuyos modelos se plantean a continuación.

III. Modelos anteriores de hipercomputación

III-A. *Máquinas con oráculos*

La primera propuesta de una alternativa a la máquina de Turing para la resolución de problemas no computables fue una del mismísimo Alan Turing, teniendo de supervisor a Alonso Church[5], donde propone una máquina de Turing normal, que es asistida por otra que Turing nombró “Oráculo”, ya que la máquina convencional asistiría a ésta, y la misma le daría las respuestas a sus consultas [10] [11].

Hasta el momento, es imposible saber como construir una máquina así, a pesar de tener fundamento lógico y matemático, la implementación se hace muy complicada, pues Turing propuso una caja negra, por lo que en la actualidad parece muy difícil obtener un avance por este camino [11]

III-B. *Máquinas de Turing acopladas*

Es un concepto acuñado por Copeland [6], que como su nombre indica son máquinas de Turing que están acopladas a un medio. Las máquinas de Turing convencionales reciben una entrada y empiezan a trabajar de manera aislada. Esta propuesta, busca que la máquina esté en constante interacción con su entorno, para recibir entradas y devolver salidas, mientras realiza una operación [1]. Se puede ver como una agregación a la teoría de Turing sobre una máquina oráculo.

III-C. *Máquinas de Turing aceleradas*

El modelo clásico de la máquina de Turing asume que cada operación es ejecutada en una unidad de tiempo finita y discreta. Sin embargo, se ha pensado en una máquina que por cada operación, multiplique por dos su velocidad [10]. Matemáticamente hablando, la máquina podría hacer infinitas operaciones en dos unidades, ya que la serie termina siendo menor a dos [1]:

$$\sum_{n=0}^{+\infty} \frac{1}{2^n} < 2$$

Obviamente esta máquina parece imposible de construir, porque de ser infinitas las operaciones, aún así, requeriría un tiempo para realizar cada operación, por lo que bajo las leyes físicas actuales, no hay medio para materializarla.

[5]Turing, Alan Mathison. ”Systems of logic based on ordinals.”Proceedings of the London mathematical society 2.1 (1939): 161-228.

[6]B.J. Copeland, The broad conception of computation, Amer. Behavioral Sci. 40 (1997) 690–716

III-D. *Otros modelos*

En [1], [12] se hace referencia a otros modelos filosóficos de adaptaciones de máquinas de Turing para lograr la hipercomputación. Pero todas radican en el mismo problema, son ideas, pero son incapaces de dar una forma de implementarlas a nivel material y físico, cajas negras, por otra parte, necesitan infinita cantidad de información en un espacio limitado, físicamente hablando nuestro medio parece indeterminista y las máquinas son deterministas. Pero sin duda alguna, la objeción más fuerte y real sobre el problema de la hipercomputación es que ninguna máquina de sistema discreto y finito puede llegar a ser hipercomputacional, por el simple hecho de que, de serlo, significa que podría ser simulada por una máquina de Turing convencional, al hablarse de un conjunto de operaciones finitas [1].

Todos los problemas que se definieron, demuestra que la hipercomputación en su principios,era más una crítica al modelo computacional establecido, que una realidad tangible.

IV. Nuevos paradigmas de la hipercomputaciónl

Como los modelos teóricos planteados no toman en cuenta la futura implementación, se decidió cambiar el paradigma de un análisis matemático y lógico a uno físico y estadístico donde la inteligencia artificial y la computación cuántica representan grandes avances en el área [13]. Hay que destacar que, en ambos casos, los resultados son probabilísticos, por lo que se sacrifica precisión en resultados por computabilidad en tiempo razonable.

IV-A. *Computación cuántica*

La computación cuántica ha adquirido un protagonismo polémico en la hipercomputación, al rechazarse por un malentendido conceptual sobre la computación cuántica estándar y la computación cuántica universal propuesta por David Deutsch [13]. David Deutsch propuso un modelo de máquina de Turing cuántico universal, por lo que incorrectamente se piensa que es una máquina de Turing, cuando en realidad es capaz de correr algoritmos cuánticos [14], de los cuales ya hay existencia de algunos que demuestran ser hipercomputacionales [13].

La computación cuántica intervienen las leyes de la mecánica cuántica, trabaja con qubits, en vez de bits, esto le permite por medio de la superposición de estados computar varios estados ortogonales de una partícula subatómica al mismo tiempo, hasta que se realiza una medición y colapsa en la salida final [14]. Según la misma fuente:

> El número de qubits indica la cantidad de bits que pueden estar en superposición. Con los bits convencionales si teníamos un registro de tres bits había ocho valores posibles, y el registro sólo podía tomar uno de esos valores. En cambio, si tenemos un vector de tres qubits, la partícula puede tomar ocho valores distintos a la vez gracias a la superposición cuántica.

Como se puede ver, al evaluar una cantidad de estados que crece de manera exponencial en el tiempo, se maneja algo parecido a lo que en el mundo clásico sería una máquina de Turing acelerada.

El modelo cuántico ha suspuesto un avance importante en el área gracias a la invención de algotimos cuánticos no computables por MT, como el algoritmo de Kieu, del que se habla en [13]

Una crítica que recibe el modelo (además de las que soportan que no se puede hablar de supercomputación en cuántica) y que es probabilista, es que la incoherencia cuántica aumenta a medida que se retrasan las mediciones, pero al medir, se pierden las propiedades cuánticas, por lo que, si se quiere precisión, el modelo no rinde tan bien, pero si se quiere que rinda, no es tan preciso. Además, la misma incoherencia, hace muy difícil recuperar los estados anteriores que llevaron al resultado, por lo que en esencia, la máquina es una caja negra muy difícil de estudiar [14].

IV-B. Inteligencia artificial

El ser humano siempre ha estado en búsqueda de imitar las capacidades del intelecto, las ideas y pensamientos. A una máquina de Turing se opondría una red neural, existiendo alguna posibilidad de que estas últimas fueran capaces de hacer algo que queda más allá del alcance de las primeras, pero por ahora, no es así [4].

Sin embargo, las redes neurales, se pueden comportar de distinta forma a partir de su arquitectura y los pesos establecidos. Existe una arquitectura donde las neuronas se autopropagan su propio error, siendo recurrentes". Las redes neurales recurrentes bajo una función de activación adecuada y unos pesos restringidos a números enteros, equivale a un autómata de estado finito, mientras que si se admiten números racionales puede adquirir las destrezas de una MT, pero si se usan números reales entre los pesos de las neuronas, son capaces de resolver problemas no computables de manera probabilísticamente acertada y en un tiempo razonable, tratando problemas incomputables y dando indicios de un modelo de hipercomputación [10].

Tiene sentido que un modelo cercano al cerebro rompa con lo que equivale a una máquina de Turing, ya que el cerebro humano parece trabajar diferente a una MT, lo raro es que precisamente son máquinas basadas en MT las que corren esos algoritmos, llegando a tener capacidades que se pueden considerar cognitivas para el ser humano [4].

Una de las quejas contra este modelo, es que pese a que en papel sea un sistema continuo, al aplicarse sobre modelos basados en MT, en el fondo resultan ser discretos, por lo que el modelo como tal a pesar de entrar dentro de la hipercomputación, en la práctica sigue estando regido por la teoría actual [4].

V. Conclusiones

Se está entrando a una nueva era de la hipercomputación y dejando atrás la idea de la computación clásica para la resolución de problemas no computables por máquinas de Turing. Todo esto, debido al cambio de paradigma que se vino dando sobre el tema en los últimos años, pues se decidió empezar a pensar en modelos funcionales basados en la física y la probabilidad, que los modelos teóricos, matemáticos y lógicos que pese a ser exactos, su implementación es muy complicada o de lleno imposible con los conocimientos de física y materiales actual.

Hoy, ya es poco común hablar de problemas no computables, sino que se refieren a éstos como problemas no computables por una máquina de Turing, ya que el movimiento relativista y funcionalista han dado sugerencias de posibles métodos de resolución de estos problemas .

Los nuevos modelos, tienen en su contra, que no son exactos o 100 % precisos, por lo que no siempre son capaces de resolver los problemas. Esto hace que su credibilidad se vea afectada al momento de hablar de computabilidad de verdad. Sin embargo, si se toma el término como brindar la solución a una función, son capaces de lograrlo y por lo tanto, con pequeñas mejorías pueden desestimar sus puntos en contra.

En el caso de demostrarse que todos los problemas son computables con una máquina de Turing, perdería el sentido hablar de hipercomputación, pues todos los problemas entrarían en el mismo conjunto[7].

Referencias

[1] Copeland, B. Jack. "Hypercomputation: philosophical issues."Theoretical Computer Science 317.1-3 (2004): 251-267.
[2] Baldor, Aurelio. "Álgebra."2009.
[3] Wikipedia, Wvbailey: "Function machine". 2020 [En línea]. Disponible en: https://en.wikipedia.org/w/index.php?title=File:Function_machine.png [13- Mayo- 2020]
[4] Becerra, José Francisco Villalpando. .Ánálisis asintótico con aplicación de funciones de Landau como método de comprobación de eficiencia en algoritmos computacionales..ᵉ-Gnosis 1 (2003).
[5] Alonso, Enrique. "De la Computabilidad a la Hipercomputación..ᴬzafea: revista de filosofía 8 (2006): 121-146.
[6] Copeland, B. Jack. "The church-turing thesis."(1997).
[7] ResearchGate, Alessandro E. P. Villa: .Án interactive Turing machine with advice". 2019 [En línea]. Disponible en: https://www.researchgate.net/figure/An-interactive-Turing-machine-with-advice_fig2_272684665 [13- Mayo- 2020]
[8] Alfonseca, Manuel. "La máquina de Turing."Números, Las matemáticas del siglo XX una mirada en 101 artículos (2000).
[9] Copeland, B. Jack. "Hypercomputation."Minds and machines 12.4 (2002): 461-502.
[10] Sicard, Andrés, and Mario Vélez. "Hipercomputación: la próxima generación de la computación teórica.Revista Universidad EAFIT 123 (2001): 47-51.
[11] Copeland, B. Jack. "Super turing-machines.Çomplexity 4.1 (1998): 30-32.
[12] Barchini, Graciela Elisa. "Las máquinas de Turing como modelo general de la computación.¿ Hacia un cambio de paradigma?.Revista Colombiana de Computación 10.1 (2009): 7-25.
[13] Sicard, Andrés, Juan David Suárez Ospina, and Mario Vélez. "Hipercomputación desde la computación cuántica.Revista Colombiana de Computación 7.2 (2006): 66-82.
[14] Bonillo, Vicente Moret. "Principios fundamentales de computación cuántica.Üniversidad de La Coruña (2013).

[7] Véase el problema P=NP