

¿Son las computadoras agentes inteligentes capaces de conocimiento?*

Are computers intelligent agents capable of knowledge?

GUSTAVO ESPARZA**

DANIEL MARTÍNEZ***

Resumen. El objetivo del presente artículo es estudiar los fundamentos filosóficos de la arquitectura de programación en dos sistemas de Inteligencia Artificial (*AlphaGo* y *Hide and Seek*). El problema dilucida la distinción epistemológica de los conceptos “conocimiento”, “intuición” y “abducción”, para definir si el cumplimiento exitoso de una métrica programada, por parte de una computadora, es condición suficiente para atribuirle un comportamiento inteligente. A través del análisis de ambos ejemplos se muestran dos cuestiones: i) el cumplimiento exitoso de un objetivo programado ofrece nuevos recursos de conocimiento, ii) dichos conocimientos dependen de la ejecución de un programa cuyo procesamiento es desarrollado por una IA y, por tanto, las operaciones superan las capacidades intelectivas humanas. Las conclusiones apuntan a que las computadoras son recursos de conocimiento especiales de comprobación de hipótesis.

Palabras clave: Inteligencia Artificial, Inteligencia Humana, abducción, AlphaGo, Hide and Seek.

Abstract. The aim of this paper is to study the philosophical foundations of the programming architecture in two Artificial Intelligence systems (*AlphaGo* and *Hide and Seek*). The problem elucidates the epistemological distinction of the concepts “knowledge,” “intuition” and “abduction,” in order to define whether the successful fulfillment of a programmed metric, by a computer, is a sufficient condition to attribute intelligent behavior to it. Through the analysis of both examples, two issues are shown: i) the successful fulfillment of a programmed objective offers new knowledge resources, ii) such knowledge depends on the execution of a program whose processing is developed by an AI and, therefore, the operations exceed human intellectual capabilities. The conclusions point to the fact that computers are special hypothesis-testing knowledge resources.

Keywords: Artificial Intelligence, Intuitive Thinking, Creative Thinking, Alpha Go, Hide and Seek.

Recibido: 15/02/2023. Aceptado: 17/06/2023.

* Agradecemos la lectura y recomendaciones de dos lectores anónimos cuyas aportaciones y sugerencias permitieron una mejora sustancial al presente trabajo. Ambos autores fuimos responsables de: planteamiento del problema, revisión bibliográfica, redacción del artículo, discusión y revisión de los avances, presentación de resultados. El orden de los autores se eligió de acuerdo al siguiente criterio: primer autor se encargaría de las gestiones con la revista (autor de correspondencia), segundo autor, daría seguimiento a las comunicaciones derivadas de lo primero.

** Universidad Panamericana, Aguascalientes, México. Profesor investigador del Instituto de Humanidades. Línea de investigación: “Kant y la tradición kantiana. Las preguntas de la sabiduría humana”, con especial énfasis en la filosofía de Cassirer. Su último libro en coautoría se titula: *The Bounds of Myth. The Logical Path from Action to Knowledge* (Brill, 2021). Miembro del Sistema Nacional de Investigadores, Nivel 1.

*** Centro de Investigaciones en Óptica, A.C., Aguascalientes, México. Estudiante de doctorado en el área de visión e inteligencia artificial. Su trabajo de doctorado se enfoca en el desarrollo de un sistema de navegación autónomo basado en aprendizaje reforzado profundo.

1. Introducción

El objetivo del presente artículo es estudiar los fundamentos filosóficos de la arquitectura de programación mediante la cual una máquina procesa información aleatoriamente con el fin de lograr un resultado que sobrepasa las capacidades humanas; actualmente existen computadoras cuyos resultados son considerados altamente especializados o incluso superiores a los alcanzados por cualquier ser humano. Como ejemplo de esto, consideramos la programación de *AlphaGo* (2021) como el primer agente (sistema inteligente) capaz de vencer a un campeón mundial de ‘Go’ y a las mejoras estratégicas alcanzadas en la interacción de equipos de agentes virtuales que juegan a las escondidas (en adelante *Hide and Seek*) (Baker, *et al.*, 2020). En ambos casos se mostrará a la arquitectura como un modelo de resolución de problemas cuyos resultados podrían clasificarse como una actividad cognitiva altamente compleja. Lo relevante de esto, como se mostrará más adelante, es que el éxito del programa no puede explicarse recurriendo únicamente a la capacidad cognitiva del programador, pues sus propias habilidades no alcanzarían el mismo éxito que el programa ofrece, por lo que el logro puede atribuirse a la operación del programa. Sin embargo, el mismo resultado requiere del acto de programación a cargo de un agente humano para que tal finalidad se cumpla, generándose así una interdependencia.

Antes de comenzar a indagar en este problema, establecemos tres cuestiones a tomar en cuenta: primero, si bien las posibilidades de ejecución de un programa dependen de la métrica definida inicialmente por el algoritmo diseñado, así como del algoritmo de aprendizaje en lo general, los resultados alcanzados por la computadora no pueden atribuirse al agente programador porque en la definición inicial de métricas no se detallan los procedimientos para alcanzar los resultados. Segundo, el conjunto de resultados alcanzados por la computadora depende de la definición paramétrica inicial; existe evidencia de programas (*AlphaGo* y *Hide and Seek*) que permite sostener que alguna forma de procesamiento de datos no ha sido definida inicialmente, de tal modo que la diversificación de los comportamientos y procesamiento de la información depende de la arquitectura de programación. A pesar de lo anterior, no hablaremos de “libertad artificial” o “intención artificial” atribuible a la computadora, pero, según veremos, es notable que las estrategias de juego presentados en ciertos programas no se encuentre definida de facto en los parámetros de funcionamiento y ni siquiera dentro de las estrategias tradicionales de juego desarrollados por los profesionales de dichos juegos; consideramos que este fenómeno merece una atención filosófica para aproximar una respuesta a qué ocurre tanto en el juego mismo como en la arquitectura de programación. Tercero, derivado de lo anterior, consideramos que una línea de interpretación que puede explicar estos logros es la siguiente: la definición de los parámetros y su arquitectura, al tener como intención el éxito de la tarea designada, asume el proceso de aprendizaje profundo (*Deep Learning*) como una fase de interconectividad de los datos en la cual la combinación y permutación iterada de los datos sigue (a) un patrón predefinido, pero también (b) un esquema creativo cuyo único proceso de orden es, precisamente, el parámetro de éxito definido. Esto anterior (c) implica, por un lado, que el agente humano concibe un medio para resolver un problema inicialmente planteado, pero, por el otro, requiere de una IA para ejecutar la solución. Esta es la paradoja general a la que queremos referirnos, ya que el

resultado final si bien puede presentarse como una novedad estratégica del juego, es un logro que no puede atribuirse a un ser humano directamente, pero tampoco a una computadora únicamente. Consideramos, por ende, que esta interrelación merece un estudio detallado.

El plan de trabajo queda como sigue: el segundo apartado estudia los conceptos de conocimiento, intuición y abducción; la finalidad es ofrecer un marco epistemológico interpretativo que permita definir qué acciones cognoscitivas pueden ser consideradas para explicar casos de éxito logrados por un agente inteligente y si es que alguno de los escenarios puede aplicarse al programador o una computadora. El tercer apartado describe la arquitectura de programación de ambos agentes y, además, se reflexionan sus implicaciones filosóficas apoyados de los conceptos previamente propuestos. El cuarto apartado plantea que los sistemas inteligentes construyen nuevas formas de resolución de problemas a partir de las funciones generales propuestas, lo que puede ser considerado como una solución inteligente. Finalmente, se hará notar que los objetivos alcanzados por las computadoras ofrecen nuevos marcos estratégicos de solución de problemas que regularmente no pueden ser alcanzados mediante recursos humanos, lo que permite reinterpretar tanto nuestros alcances y capacidades intelectivas.

2. Conocimiento, intuición y abducción

En el presente apartado definiremos tres conceptos epistemológicos con el fin de ofrecer un marco interpretativo que nos permita identificar qué tipo de tareas pueden atribuirse a una inteligencia humana y cuáles a una inteligencia artificial. Consideramos que la diferenciación y criterios que a continuación presentamos nos permitirá delimitar en qué medida el cumplimiento exitoso de un parámetro predefinido es susceptible de entenderse como una prueba de un agente inteligente o simplemente la reorganización de datos cuyo estudio posterior garantiza nuevas posibilidades de interpretación.

Comenzamos por definir el concepto de conocimiento partiendo de la definición clásica dada por Platón (2015) *S* conoce que *P* cuando posee una *creencia, verdadera y justificada* (Teeteto, 201d). Encontramos una ampliación de estos criterios en la defensa de Sócrates respecto de sus actividades públicas, ya que argumenta que todo sujeto debe ser capaz de argumentar cómo es posible definir un conocimiento como propio (Apología, 19d-33c). En contraparte, esta definición ha sido puesta en duda por Gettier (1963, 121-123) a partir de la elaboración de contraejemplos en los que cuestiona la suficiencia tripartita para definir que *S* conoce que *P*, lo que hace pensar que dicha definición está incompleta.

John Ian K. Boongalig (2021, 87-111) ha evaluado el cuestionamiento de Gettier en contra de la definición clásica diciendo que el argumento más fuerte propuesto por el filósofo norteamericano frente a la tesis platónica son las circunstancias azarosas que parecen justificar que *S* no conoce verdadera y justificadamente que *P*. En ese sentido Boongalig apunta que aun cuando algunos eventos particulares pueden explicarse a partir de circunstancias fortuitas, ello no permite sostener universalmente que *S* no conozca que *P*; en contraparte, el propio Boongalig considera que para sostener la propuesta de Gettier como contra argumento válido de la definición clásica, es necesario que un segundo agente (S_2) presente pruebas que demuestren categóricamente que efectivamente *S* no conoce que *P*, pues el azar mismo

no es prueba fehaciente de que S desconozca que P . Sin embargo, aún en el caso que algún agente S_n demuestre que S no conoce, el mismo criterio se aplica para la justificación de S_2 , pues un tercer agente S_3 puede demostrar que lo dicho por S_2 es erróneo. El círculo, como se aprecia, se vuelve infinito, pues en cada nuevo escenario S_n puede dudar de su antecesor y de las pruebas que éste presente, a menos que se establezca un criterio general a partir del cual determinar la imposibilidad de la duda para cualquier caso. La propuesta de Boongalig, entonces, es recolocar la definición de conocimiento en sus términos clásicos ¿puede S justificar la verdad de su creencia (P)?

Esta discusión previa es importante pues podemos reformularla de la siguiente forma para aplicarla a nuestro tema, ¿puede una IA conocer que P ? Es decir, ¿puede una IA justificar la verdad de su creencia? Como se verá más adelante con los ejemplos, una respuesta a estas preguntas puede formularse del modo siguiente:

- a. *Verdad*. Una IA otorga los resultados para los que fue programada, por ende, se puede establecer que este rubro se cumple.
- b. *Justificación*. Una IA, sin embargo, al estar programada para cumplir con una tarea específica, no puede dar una explicación convincente de cómo es que alcanzó el resultado logrado. En todo caso, dicha labor corresponde al programador.
- c. *Creencia*. Una IA opera un comando como parte de su funcionamiento. No es posible atribuir un estado mental a una computadora diciendo que ésta cree que la programación asignada es el mejor medio para resolver una métrica preestablecida. En todo caso, tal situación corresponde al agente humano quien cree que la programación puede alcanzar el parámetro definido, por ende, este rubro no es atribuible a una máquina.

En resumen, una IA no puede conocer que P , pues no cumple con los criterios previamente definidos. Si bien el programador aprovecha a la IA como instrumento para el desarrollo de nuevos conocimientos, ello no implica que la propia computadora sea capaz de conocer, sino que es un recurso para que el programador conozca.

Pasamos ahora a delimitar el concepto de intuición y para ello seguimos a Rafael Miranda-Rojas (2018) para quien, por tal, se ha de entender un “parecer intelectual” en donde el sujeto S intuye que un predicado P es verdadero, lo cual implica que “ S aprende de modo inmediato la verdad de P ”, la cual, además, “es comprendida intelectualmente, racionalmente” (Miranda-Rojas 2018, 263). Sin embargo, de lo anterior se deriva un problema general: el que S intuya que P ¿necesariamente hace que P sea verdadero? Miranda-Rojas sostiene que “no es a través de la intuición que se justifica en última instancia la verdad de cierto enunciado, el nexo causal entre que P parezca verdadero y que P sea verdadero no es un caso de infalibilidad” (Miranda-Rojas 2018, 264). De acuerdo a esto, entonces, advertimos un modo de conocimiento no discursivo, pero cuyo objeto es el conocimiento de la verdad de P , pero en cuyo proceso de fundamentación la intuición únicamente opera como un caso donde la identidad entre el sujeto y el objeto se funda en la *creencia* de que P es verdad.

Es cierto que la fundamentación de P en un acto de creencia, como es bien sabido, no constituye un conocimiento *per se*, aunque sí implica uno de sus elementos para determinar la veracidad del postulado. En ese sentido, para Robert Audi (2019) la intuición sería:

a non-inferential knowledge or grasp, as of a proposition, concept, or entity, that is not based on perception, memory, or introspection; also, the capacity in virtue of which such cognition is possible (p. 527).

En este contexto, el intercambio entre la ‘creencia’ de que P es verdadera por su *posibilidad cognitiva* constituye una renovación en los términos de fundamentación de la intuición como un medio válido para conocer la verdad de un postulado. De la definición se resaltan el carácter “no inferencial” de un “conocimiento”; ello implicaría, por tanto, que existe un conocimiento que se logra por medios no perceptivos, memorísticos o introspectivos, de tal modo que cumplirían con las condiciones de operar como *creencia, verdadera y justificada* (Platón, *Teeteto*, 201d); dicho de otro modo, para que una intuición sea considerada como tal ésta debe apuntar necesariamente a un conocimiento que sea “verdadero”, pero a través de una justificación inferencial. La principal diferencia de la ‘intuición’ y el ‘conocimiento’, está en los medios y posibilidad de “justificación” de un postulado. Si aprovechamos estos resultados podemos preguntar ¿es posible que una IA intuya que P ?

- a. *Verdad*. Como se dijo previamente, este rubro se cumple por parte de una IA.
- b. *Justificación*. De acuerdo a lo dicho, si bien una IA no puede justificar directamente el éxito de su programación, al constituirse como el instrumento o recurso para alcanzar el éxito de un objetivo, podríamos decir que la razón por la cual un proceso se cumple es, precisamente, su aprovechamiento. En ese sentido, el sentido de la justificación que el programador pueda ofrecer depende del resultado de la IA y, por ende, forma parte de la justificación.
- c. *Creencia*. En este caso ocurre un proceso similar al previo. La IA directamente no asume una creencia, pero forma parte del marco de recursos que hace que el programador pueda asumir la creencia de que un procedimiento puede ser alcanzado. En ese sentido se puede decir que el programador puede creer en la posibilidad de éxito asumiendo el desarrollo de un algoritmo como base de dicho estado mental.

En resumen, a pesar de todas las consideraciones, no es posible establecer que una IA intuya, pues buena parte de las interpretaciones requieren una inteligencia humana para lograr los criterios.

Pasamos ahora a estudiar el concepto de abducción o hipótesis propuesto por Charles S. Peirce (1998) quien la define del siguiente modo:

The abductive suggestion comes to us like a flash. It is an act of insight, although of extremely fallible *insight*. It is true that the different elements of the hypothesis were in our minds before; but is the idea of putting together what we had never before dreamed of putting together which flashes the new suggestion before our contemplation (p. 227. Énfasis en el original).

Como se aprecia, el concepto de abducción constituye un momento de conocimiento cuyo desarrollo implica un acto de “insight” o aparición. Lo relevante de este modelo es que el acto cognoscitivo no requiere de una justificación según requiere el sentido clásico del conocimiento. De acuerdo a lo que veremos en los ejemplos más adelante, todo proceso de programación se estructura como el desarrollo de una serie de proposiciones con las cuales

se espera conseguir un resultado predefinido. Ello implica que el acto de conocimiento que se alcanza no depende del acto de justificar cómo es que el proceso desarrollado por la computadora logró el resultado, sino que se sostiene de la hipótesis propuesta por el agente al momento de proponer su programa como un algoritmo capaz de resolver una tarea.

En este proceso es evidente que el acto de confirmación no implica una justificación de parte del programador mismo, sino únicamente la validación de que los comandos propuestos son un recurso efectivo para lograr el éxito establecido. Kyrylo Medianovsky y Ahti-Veikko Pietarinen (2021) insisten en que un sistema inteligente que procesa grandes cantidades de datos opera como un sistema abductivo, lo cual permite a un agente humano interpretar la información para con ello completar el conocimiento. Además de lo anterior, hacen notar que mientras que un conocimiento requiere de explicaciones causales, el procesamiento de los datos por parte de una IA únicamente requiere de una programación y un criterio que debe ser alcanzado. Esta diferencia permite reconocer que mientras que un agente humano es capaz de producir conocimiento, una computadora no puede cumplir con los mismos criterios epistémicos, sin embargo, ello no implica que los resultados entregados carezcan de validez para el agente humano que los interpreta.

Los autores proponen varios ejemplos para mostrar, por un lado, que actualmente una computadora por sí sola no puede detectar el grado de riesgo patológico, meteorológico o sísmico, de los datos que procesa, pero sí, en cambio ofrecer nuevos modelos de organización de datos ofreciendo datos cuya interpretación no podría alcanzarse únicamente aprovechando las capacidades humanas. Medianovsky y Pietarinen (2021) remarcan que es importante aceptar el lugar que ocupan las IA dentro de la producción de conocimiento.

De acuerdo con esto, se puede entender mejor el lugar que desempeña una IA en el esquema general hasta aquí planteado. Al operar como un sistema operativo que favorece la producción y desarrollo de conocimiento, el mejor esquema explicativo tiene que ser uno que reconozca las funciones abductivas que una computadora ofrece a un agente humano para lograr sus tareas y actividades de investigación.

Colocamos una última cuestión previo al análisis de los programas *AlphaGo* y *Hide and Seek*. John McCarthy, a mediados de la década de los 50's, convoca a una reunión de investigación en Darmouth para delimitar tanto los problemas propios de una *AI*, como su campo de investigación. Una de las cuestiones centrales que proponía estudiar era la aleatoriedad y la creatividad (*Randomness and Creativity*), cuya conjetura general proponía: "The randomness must be guided by intuition to be efficient" (McCarthy, et al., 1955, 2). Con dicha hipótesis el autor configuraría el campo de la inteligencia artificial, del cual más tarde se derivarían ramas como el aprendizaje profundo (*Deep Learning*) y aprendizaje por reforzamiento (*Reinforcement Learning*), y que hoy en día ofrecen una gran cantidad de evidencia computacional mostrando dos cuestiones importantes:

- 1) Un sistema inteligente basado en aprendizaje por reforzamiento opera en función de la delimitación de una métrica que define el éxito o fracaso de un resultado.
- 2) A pesar de que se delinea una arquitectura de funcionamiento, el procesamiento de la información no depende de la estructuración de los comandos iniciales, sino de la métrica general estipulada, lo que implica que el éxito está en función de la cantidad y tipo de datos que el programa puede combinar para resolver un problema.

A partir de esto, planteamos que existen programas basados en *Deep Reinforcement Learning* cuya operación cumple con las condiciones del concepto de abducción previamente expuesto, pues el agente programador establece una idea general que luego es alcanzada por un sistema de procesamiento de la información que le permite lograr una métrica general predefinida. Para demostrar esto, en el siguiente apartado se describirá la arquitectura de funcionamiento de dos programas cuyos resultados cumplen con las especificaciones métricas de sus diseñadores, sin que por ello implique que tales logros sean atribuibles o replicables por ellos mismos; en el caso de *AlphaGo* no es posible sostener que los programadores puedan ser considerados “campeones mundiales del juego” y, en el caso de *Hide and Seek*, los esquemas de búsqueda o escondite estén ya estipulados en el comando, sino que se derivan de las métricas y arquitectura del programa en sí, por lo que el procesamiento de la información no está supeditado a la conformación de estrategias de éxito del programa, sino al cumplimiento de la meta definida como final.

3. *Alpha Go* y *Hide and Seek*. Arquitecturas de programación

Dentro de las distintas propuestas de IA encontramos modelos procesamiento de información, así como de validación de éxito de una tarea previamente definida, en donde el único criterio de comprobación es la satisfacción de la propia métrica predefinida. En este sentido, se dice que una programación ha cumplido con su función si desarrolla la tarea asignada; analógicamente diríamos que un programa es exitoso si cumple con las tareas definidas inicialmente por el programador, lo cual implica que la verdad del programa (su éxito) está predefinido por el propio agente.

A continuación, presentamos dos ejemplos de IA en los que se diferenciará entre la programación que propone un agente humano para cumplir con ciertas tareas que él mismo no puede cumplir con recursos y capacidades humanas, por lo que es necesario recurrir al procesamiento de información a través de una computadora inteligente, la cual, si bien responde a criterios de programación pre establecidos, sus resultados se derivan de las funciones y capacidades tecnológicas de la computadora misma. Veamos ambos casos con detalle.

3.1. *Mastering the Game*. La intuición como métrica de programación

Un ejemplo de IA cuyo objetivo fue alcanzar una habilidad “súper humana” en el juego de ‘Go’, es el programa de *DeepMind*, *AlphaGo* y sus diferentes versiones. Algo importante a resaltar de *AlphaGo* es que fue la primera inteligencia artificial en derrotar a jugadores profesionales de Go, incluyendo al 18 veces campeón mundial, Lee Sedol (Granter, Beck y Papke Jr., 2017, 619-661). El proceso de entrenamiento de *AlphaGo* se detalla a continuación.

En la primera fase de entrenamiento se utilizó *Aprendizaje supervisado* (*Supervised Learning*). Inicialmente se entrenó a *AlphaGo* con cientos de jugadas de jugadores profesionales para que la IA “aprendiera” las reglas del juego “observando” las jugadas estratégicas convencionales (Gibney, 2016, 445-446). De acuerdo con Florian Brunner (2019):

This network takes the current board state as a 19x19x48 batch as input and outputs a probability distribution over all legal moves a . The network has been trained on randomly sampled state-action pairs (s,a) , using stochastic gradient ascent to maximize the log likelihood of selecting move a in state s [...] This network predicted expert moves with an accuracy of 57.0% (p. 7).

Con el uso de redes neuronales profundas para calcular las probabilidades de movimientos, lo que se pretende es garantizar una toma de decisiones basada en los cálculos que la propia computadora puede procesar, pero, al mismo tiempo, ordenar dichas operaciones a una métrica general determinada. De este modo, se aprecia que, si bien la delimitación del evento exitoso y las fórmulas para lograrlo son precisadas por los programadores, no implica que esté definido el medio a través del cual es posible lograr el objetivo.

En la segunda fase, una vez lograda la tasa de éxito esperada, se entrenó a *AlphaGo* utilizando *Aprendizaje reforzado (Reinforcement Learning)*. El objetivo general del entrenamiento fue recalibrado para, en lugar de predecir las jugadas y movimientos del experto, ahora se fijaría como meta el “ganar los juegos”. En este estadio, la IA simuló millones de partidas contra sí misma con el fin de evaluar la tasa de éxito de las estrategias convencionales, además de desarrollar nuevas opciones de juego para garantizar el cumplimiento la métrica general. David Silver (2017) y su equipo al respecto aclaran: “The policy network was trained initially by supervised learning to accurately predict human expert moves, and was subsequently refined by policy-gradient reinforcement learning. The value network was trained to predict the winner of games played by the policy network against itself” (354-359). Con este reajuste en la lógica de operación, el programa, si bien calculaba los posibles movimientos del oponente para generar una estrategia de juego acorde a la situación, su métrica ahora implicaba que, en un segundo momento, debía delinear nuevas estrategias para alcanzar el éxito el cual se definía como “ganar” el juego.

Eventualmente *AlphaGo Zero* vencería en una serie de juegos a su predecesora *AlphaGo* con un resultado de 100 - 0. Esto permitió sostener que esta IA tiene un mejor desempeño en el juego de Go que su versión previa. David Silver, uno de los investigadores principales del proyecto, afirma que el aprendizaje inicial de *AlphaGo* a partir de las jugadas de profesionales humanos, es lo que limita la creatividad de esta IA, sesgando sus estrategias hacía la experiencia humana. Es precisamente el hecho de que *AlphaGo Zero* aprende lo que se traduce en la exploración del juego desde una perspectiva única¹. Pero ¿cómo es que aprende esta inteligencia artificial? A continuación, revisaremos el funcionamiento general de la arquitectura de *AlphaGo Zero*.

1 Al respecto, son interesantes las reflexiones de Granter, Beck y Papke Jr. (2017) pues muestran tanto áreas que superan las capacidades humanas como sus propias limitaciones al respecto: “AlphaGo’s success at surpassing human experts, computer vision algorithms fall short of matching some basic human vision capabilities. When humans look at pictures, they can interpret scenes and predict within seconds what is likely to happen after the picture is taken (“object dynamic prediction” in computer terminology). However, although algorithms can be trained to make similar predictions in abstract cartoon scenes, they cannot accurately predict what will happen next in real-world photographic scenes. And algorithms are notoriously bad at some aspects of image analysis. For example, algorithms cannot accurately predict whether a human will find a photograph funny or not, to the point where humor detection is considered an “AI-complete problem””. (p. 620).

Esta IA está compuesta por tres etapas principales. Una *primera red neuronal profunda* para predecir el posible ganador de la partida en cada estado del juego. Es decir, a partir del estado actual del tablero, la red entregará como salida valores entre '1' y '-1', de modo que, si hay una alta posibilidad de que el jugador 1 gane, el valor de salida se aproximará a 1, pero si el jugador 2 tiene mayor posibilidad de ganar la partida, el valor se aproximará a -1, y si el juego tiene alta posibilidad de terminar en empate, el valor de la red se encontrará cercano a 0 (Brunner, 2019, 8-9). Esta red neuronal permite evaluar el estado actual del juego en cada movimiento y la confianza y exactitud de la predicción de esta red depende de la experiencia obtenida por la IA (conocimiento de un mayor número de estados del juego).

Una *segunda red neuronal profunda* para determinar el mejor movimiento a partir del estado del juego. Esta red, "observa" el tablero y los últimos movimientos de ambos jugadores para determinar cuál es el siguiente movimiento que el jugador debe hacer. Como salida, la red neuronal entrega una matriz de valores entre '0' y '1' para cada posición en el tablero. La posición con el mayor valor es la mejor jugada (conocida) para ese estado del tablero. La optimización de esta red se da a través de evaluar si el movimiento fue beneficioso para el resultado final (ganar o perder la partida). De este modo la red neuronal se ajusta para sugerir mejores movimientos conforme a su experiencia.

La tercera etapa es un árbol de búsqueda de Monte Carlo (MCTS, por sus siglas en inglés. Ver figura 3.1), este algoritmo realiza internamente simulaciones de los posibles movimientos tanto de la IA como del oponente. De este modo, *AlphaGo Zero* crea un árbol ramificando diferentes jugadas y los posibles desenlaces de cada jugada. En esta etapa, el MCTS utiliza la red neuronal de la segunda etapa para predecir cuales son las mejores jugadas que la IA puede hacer, y cuáles podrían ser las mejores jugadas del oponente. Finalmente se utiliza la rama del MCTS que maximice las posibilidades de ganar (Silver, et al., 2017, 354).

El MCTS y la red neuronal de la segunda etapa se optimizan utilizando referencias circulares, es decir, la red neuronal se ajusta comparando sus predicciones con el MCTS, y a su vez, el MCTS utiliza las predicciones de la red neuronal para simular las mejores jugadas posibles. De este modo, conforme el MCTS explora más jugadas, la red neuronal puede intuir de mejor manera las mejores jugadas, y producir mejores resultados en las simulaciones del MCTS, y así sucesivamente (Silver, et al., 2017, 355-356). Luego de millones de juegos de experiencia, *AlphaGo Zero* produce jugadas infalibles a través de una intuición "súper humana" del juego (Silver, et al., 2017, 358).

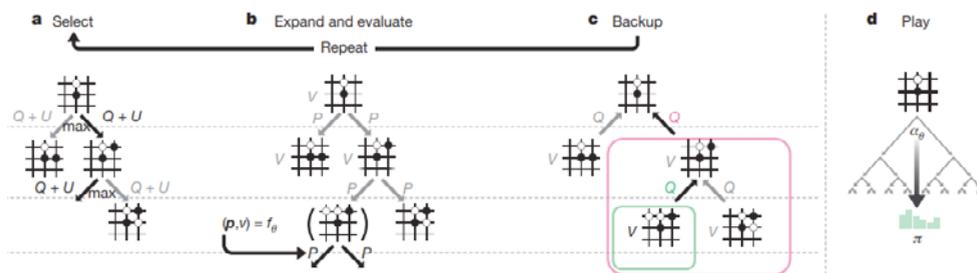


Figura 3.1 Árbol de búsqueda de Monte Carlo para Alpha Go (Silver, et al., 2017, 355.)

La parte creativa de *AlphaGo Zero* se produce a través de un parámetro del MCTS que permite a la IA explorar nuevas jugadas, incluso si estas no se ajustan a las estrategias convencionales del juego desarrollado por profesionales, esta exploración aleatoria lleva a *AlphaGo Zero* a desarrollar jugadas nuevas y evaluar si estas se aproximan a un escenario de éxito (1) o fracaso (-1). Durante la etapa de entrenamiento, este parámetro es más alto para maximizar la exploración de *AlphaGo Zero*, mientras que en partidas competitivas es menor, lo cual lleva a la IA a ejecutar las mejores jugadas previstas de acuerdo con su experiencia; sin embargo, permanece una pequeña posibilidad de hacer una jugada “creativa” o aleatoria (Silver, et al., 2017, 360-361). Lo que *AlphaGo Zero* demuestra es que puede desarrollar un estilo de juego basándose en los datos almacenados luego de jugar contra sí mismo. Al respecto, vale la pena reproducir las conclusiones de David Silver (2017) y su equipo sobre el logro alcanzado por su programa:

Humankind has accumulated Go knowledge from millions of games played over thousands of years, collectively distilled into patterns, proverbs and books. In the space of a few days, starting *tabula rasa*, AlphaGo Zero was able to rediscover much of this Go knowledge, as well as novel strategies that provide new insights into the oldest of games (p. 358).

No deja de ser provocador este planteamiento, pues dicha afirmación se basa en los resultados logrados por una computadora enfrentando al campeón mundial del juego de Go. Pero lo que a nosotros nos interesa de estas conclusiones son dos cuestiones: (a) que el equipo de Silver, digamos, la parte humana, únicamente programó un proceso de juego al definir la métrica, los métodos de recuperación de información, así como el procesamiento de los datos, pero no así las estrategias para “ganar”; (b) al establecer como rango de desarrollo desde la *Tabula rasa* hasta la victoria contra Lee, implica que los programadores únicamente consideraban que a través de la programación de un sistema sería posible alcanzar la victoria, pero aún con ello no es posible sostener que fuesen capaces de derrotar directamente a Lee; es decir, para el caso de David Silver y su equipo, a pesar de su capacidad para programar una computadora que se alzó la victoria en contra del campeón de Go, ellos mismos no son capaces de genera una estrategia ‘humana’ que emule el mismo evento.

3.2. *Hide and Seek. La creatividad como estrategia de intuición*

Open AI (2019) ha desarrollado también diversas aplicaciones con inteligencias artificiales que han demostrado un comportamiento intuitivo, creativo y adaptable. Entre ellos se encuentra *Hide and Seek*, esta aplicación consiste en una serie de simulaciones en diferentes entornos virtuales donde dos equipos de agentes (escondidos y buscadores) aprenden el juego de “las escondidas” (Baker, et. al., 2019). El objetivo de este juego es que los participantes involucrados generen nuevas estrategias de éxito vinculadas al rol que desempeñan (esconderse o encontrar), lo que ha implicado un buen parámetro de medición para los programadores en cuanto al desarrollo de estrategias y de creación de acción. De acuerdo con Baker y su equipo, las reglas que tomaron en cuenta para su programa fueron las siguientes:

We introduce a new mixed competitive and cooperative physics-based environment in which agents compete in a simple game of hide-and-seek. Through only a visibility-based reward function and competition, agents learn many emergent skills and strategies including collaborative tool use, where agents intentionally change their environment to suit their needs (Baker, et. al., 2019, 3).

Se esperaba que los agentes cumplieran con una métrica definida y designada para cada participante (esconderse o encontrar), a través de una toma de decisiones, la cual, a su vez, dependía de una red neuronal profunda (Ver figura 3.2) para procesar información como: (1) La posición y velocidad tanto de sí mismo, como la de los otros agentes; (2) Campo de visión cónico; (3) La posición, velocidad y distancia de objetos del entorno como cajas, rampas o bloques. Mientras que, por otro lado, la salida de la red neuronal es la siguiente acción del agente, las acciones permitidas son: (i) Movimiento en cada dirección; (ii) Voltar a los alrededores; (iii) Sujetar objeto (para desplazarlo).

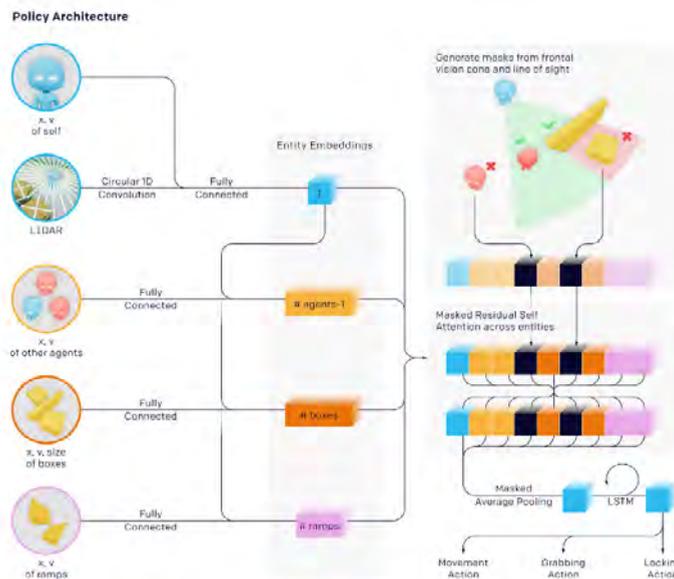


Figura 3.2 Arquitectura de agentes de Hide and Seek (Baker, et. al., 2019)

A partir de este marco, el comportamiento de los agentes está regido por una sola métrica cuyas políticas de recompensa pueden ser: (i) todo el equipo de buscadores es recompensado positivamente si un buscador encuentra a un agente escondido o recompensado negativamente si el equipo no puede encontrar a ningún escondido: (ii) el equipo de escondidos sea recompensado positivamente si logran mantenerse fuera del campo de visión de los buscadores o, también, si uno de los escondidos es encontrado todo el equipo recibe una recompensa negativa. De este modo, las simulaciones tratan de fomentar no sólo la creatividad de los

agentes para encontrar estrategias para esconderse, si no también, colaborar para que todos los miembros del equipo logren mantenerse fuera de visión.

De acuerdo con Baker y su equipo, en general los programas no existen estímulos explícitos que motiven a los agentes a interactuar con objetos del ambiente, por lo que la métrica debe asegurar que quien busca o se esconde, responda a la demanda de lograr un estímulo positivo (1) y evitar el negativo (-1), siendo, en ambos casos, definido por el sistema de programación, antes de iniciar el juego; pero ¿qué propósito tienen las recompensas en el contexto del aprendizaje de los agentes? En el aprendizaje reforzado, los agentes interactúan con el entorno a través de dos acciones básicas que eventualmente se traducen en una serie de movimientos tales como desplazar objetos de izquierda a derecha, abajo hacia arriba, o incluso empalmar más de un objeto para construir refugios (para el caso de quien se esconde) o trepar objetos, moverlos del lugar en el que se encuentran, rodear a otros agentes para evitar que escapen (para el caso de quien busca)². Al terminar la iteración, se evalúa si el agente cumplió o no el objetivo. En caso de cumplir la meta, la red neuronal del agente promueve la ejecución de esa serie de acciones que llevaron a cumplir el objetivo, si el agente fracasa, la red neuronal desalienta al agente de realizar esa serie de acciones. Al cabo de cientos, miles, o incluso millones de iteraciones (dependiendo de la complejidad del entorno), el agente es capaz de intuir como interactuar con el entorno para maximizar la recompensa.

Open AI (2019) obtuvo resultados muy interesantes con respecto a la colaboración, creatividad y adaptación de los agentes para el caso particular del *Hide and Seek*; en uno de los experimentos realizados, obtuvieron los siguientes sucesos (ver figura 3.3):

- a. Los agentes actuaban aleatoriamente, ya que no comprendían el objetivo del juego.
- b. Los buscadores aprendieron a buscar a los escondidos.
- c. Los escondidos aprendieron a bloquear las entradas de los buscadores utilizando cajas.
- d. Los buscadores aprendieron a usar rampas para saltar las paredes que les impedían entrar al cuarto donde se encontraba el equipo contrario.
- e. Finalmente, el equipo de escondidos aprendió a esconder las rampas y bloquear la entrada para evitar que los buscadores saltaran la pared.

2 El proceso de movimiento es detallado del modo siguiente: “Agents are simulated as spherical objects and have 3 action types that can be chosen simultaneously at each time step. They may move by setting a discretized force along their x and y axis and torque around their z -axis. They have a single binary action to grab objects, which binds the agent to the closest object while the action is enabled. Agents may also lock objects in place with a single binary action. Objects may be unlocked only by agents on the team of the agent who originally locked the object. Agents may only grab or lock objects that are in front of them and within a small radius”, (Baker, et. al., 2019, p. 4).

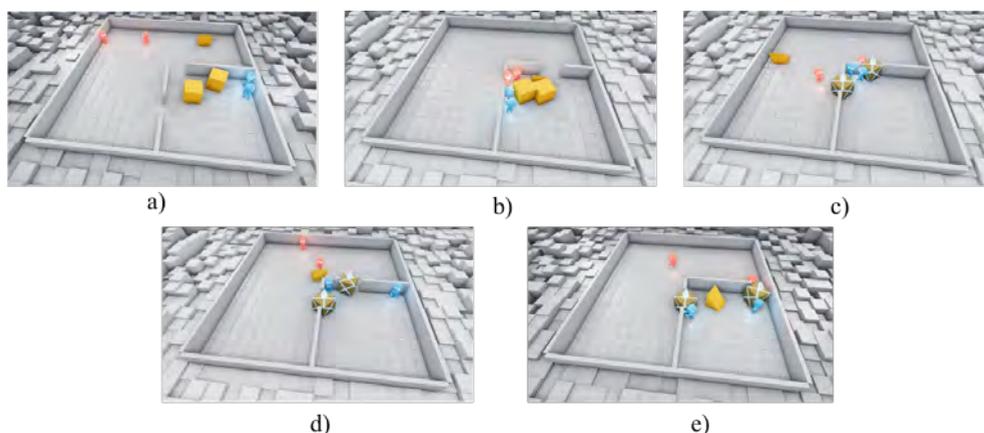


Figura 3.3 Resultados simulación *Hide and Seek* (Baker, et. al., 2019).

De esta serie de eventos, una vez que los equipos de agentes comprendieron el objetivo del juego y encontraron estrategias para cumplir la meta, generaron que el equipo contrario a adaptarse, creando nuevas estrategias que, a su vez, exigieron al equipo contrario a responder con la renovación estratégica de acciones para contrarrestar las opciones de búsqueda del equipo contrario. Este patrón se repitió continuamente hasta que el equipo de los escondidos encontró la forma de bloquear todas las posibilidades de los buscadores. Al menos para este entorno en particular, en otros entornos más complejos, se obtuvo el mismo patrón de adaptación entre ambos equipos, sin embargo, en algunas ocasiones, los agentes aprendieron incluso a aprovecharse del sistema de físicas de la simulación para crear nuevas estrategias. El propio equipo de Baker resalta estos logros como parte de una estrategia creativa cuya finalidad es asegurar la métrica final esperada:

As agents train against each other in hide-and-seek, as many as six distinct strategies emerge, each of which creates a previously non-existing pressure for agents to progress to the next stage. Note that there are no direct incentives for agents to interact with objects or to explore, but rather the emergent strategies are solely a result of the autocurriculum induced by multi-agent competition (Baker, et. al., 2019, 6).

Al resaltar el valor de generación de “nuevas” estrategias, lo que se subraya es, por un lado, que la métrica predefinida por parte del equipo de programación establecía solamente las recompensas a lograr, aspecto que era buscado por el equipo de búsqueda y de ocultamiento. Sin embargo, el objetivo del equipo de Baker era medir el proceso de desarrollo de recursos mediante los cuales ambos programas cumplirían su objetivo hasta el punto en que no fuese posible por parte del equipo contrario encontrar soluciones a un problema inicial dado, de tal modo que el desarrollo del juego implicaría que uno de los dos equipos eventualmente no podría obtener la recompensa, mientras que el otro la lograría de modo indefinido.

Conclusiones

En el presente artículo nos propusimos como objetivo estudiar los fundamentos filosóficos de la arquitectura de programación mediante la cual una máquina procesa información con el fin de lograr un resultado exitoso que derive en un principio de orden. Para ello consideramos dos ejemplos de programación (*AlphaGo Zero* y *Hide and Seek*) y con ello mostrar el proceso mediante el cual los programadores lograron el cumplimiento de métricas definidas a través de recursos computacionales.

Partimos de la delimitación y objetivos estipulados por McCarthy en la conferencia en Dartmouth, en donde se establecía que una de las tareas a desarrollar por parte de una *AI* correspondía al desarrollo de un pensamiento intuitivo como organizador de la creatividad. Consideramos los casos de *AlphaGo Zero* y *Hide and Seek* porque ambos plantean soluciones a juegos o simulaciones cuyos procesos suelen atribuirse a agentes humanos. Sin embargo, consideramos estos escenarios porque en ambos contextos se observa cómo la definición paramétrica de resultados no define el tipo de resultados por alcanzar, sino que, precisamente, se realizan para describir las estrategias mediante las cuales es posible alcanzar una meta de éxito. Los resultados obtenidos por *DeepMind* y *Open AI* –las empresas desarrolladoras de ambos programas– demuestran el potencial de la inteligencia artificial para resolver problemas en formas creativas que agentes humanos no han mostrado u ofreciendo alternativas de solución que resultan novedosas.

Como parte del análisis filosófico, se dispuso estudiar tres conceptos epistemológicos que nos permitirían evaluar el tipo de procesamiento que podía atribuirse a una *IA*. Se distinguió entre conocimiento (definido como creencia, verdadera, justificada), intuición (definido como creencia verdadera, pero que carece de justificación) e inducción o hipótesis (proceso cognitivo que requiere un caso y una regla para establecer una proposición verdadera). En este punto se estableció que los primeros dos conceptos no pueden ser vinculados con una computadora pues el conocimiento y la intuición implican operaciones que por ahora una *IA* no ha mostrado poseer y que, además, no se requieren por parte de una computadora para lograr el éxito de una métrica previamente definida.

En cambio, en el caso de la abducción, al tratarse de un proceso en donde se plantea un nuevo conocimiento, pero que no requiere de una justificación de cómo es que se obtuvo el resultado, se pudo mostrar que una actividad cognitiva de esta naturaleza permite un mejor modo de comprensión del tipo de papel que juega tanto un agente humano que aprovecha un programa computacional para resolver un problema. Con el modelo propuesto por Peirce es posible aceptar que el valor central de una *IA* no está en la capacidad reflexiva o la descripción de cada una de las fases del proceso que se cumple para lograr un resultado, sino, precisamente en la taza de logros alcanzados con un programa que demuestre ser la vía o recurso intelectual con el cual sobre pasar las capacidades intelectuales de un ser humano.

Para lograr esto anterior, emprendimos un análisis de los programas *Hide and Seek* y *Alpha Go*, concluimos que el uso e intermediación de una *AI* para la resolución de un problema del cual se desconocen las estrategias o recursos para la solvencia de la métrica, puede considerarse un caso de conocimiento intuitivo pues los equipos programadores, a pesar de que definen la meta final –delimitan el valor de éxito o verdad a la que debe llegar el programa– desconocen el proceso a través del cual alcanzar este escenario.

Con todo ello, consideramos que la reflexión sobre el papel que juegan las computadoras en la generación del conocimiento y, sobre todo, la posibilidad de justificar un resultado logrado como un proceso de búsqueda intencional y racional, validan la propuesta de McCarthy con respecto a la posibilidad que un sistema de IA basado en aprendizaje por reforzamiento opera en función de la delimitación de una métrica que define el éxito o fracaso de un resultado. Si bien los resultados ofrecidos por una computadora no pueden asumirse como conocimiento propiamente dicho, con los logros alcanzados por los programas estudiados se muestra que no toda mejora y desarrollo de nuevos recursos y medios de solución deben provenir de una inteligencia humana.

Esto último, recoloca el avance tecnológico en una nueva dimensión, pues no se puede afirmar *per se* que el vertiginoso avance de la IA constituya un atentado o problema para el ser humano, pues se trata de una herramienta que puede contribuir a la mejora de la sociedad. En todo caso, la pregunta más importante debe redirigirse a qué tipo de conocimientos buscan ser alcanzados por los agentes quienes a través de sus programas pretenden demostrar sus hipótesis. Una computadora, como se mostró, únicamente explora los mejores medios y recursos estratégicos para validar o desacreditar dicha proposición.

Referencias Bibliográficas

- AlphaGo*. (2021). *Google DeepMind. Alpha Go*. Recuperado de <https://deepmind.com/research/case-studies/alphago-the-story-so-far>.
- Audi, R. (2019). "Intuition", *The Cambridge Dictionary of Philosophy*. Tercera Edición. Cambridge: Cambridge University Press, 2019.
- Baker, B., Kanitscheider, I., Markov, T., Wu, Y., Powell, G., McGrew, B. y Mordatch, I. (2020). "Emergent tool Use from Multi-Agent Autocurricula", *The International Conference on Learning Representations* (conference paper), pp. 1-28. Disponible en: <https://arxiv.org/pdf/1909.07528.pdf>
- Boongalig, J. (2021). *Disolving the Gettier problema. Beyond analysis*. Cambridge: Cambridge Scholars.
- Brunner, F. (2019). "Mastering the game of Go with deep neural networks and tree search", *Artificial Intelligence for Games Seminar Report*, Heidelberg University, pp. 1-15. https://hci.iwr.uni-heidelberg.de/system/files/private/downloads/36349047/report_florian_brunner.pdf
- Gettier, E. (1963). "Is Justified True Belief Knowledge?" *Analysis*, Vol. 23, No. 6, pp. 121-123. <https://doi.org/10.2307/3326922>
- Gibney, E. (2016). "Google masters Go. Deep Learning software excels at complex ancient board game". *Nature*. Vol. 529, pp. 445-446. <https://www.nature.com/articles/529445a.pdf>
- Granter, S., Beck, A. y Papke Jr., D. (2017). "Alpha Go, Deep Learning, and the Future of the Human Micropist", *Arch Pathol. Lab Med*, Vol. 141, No. 5, pp. 619-61. <https://doi.org/10.5858/arpa.2016-0471-ED>
- McCarthy, J., Minsky, M., Rochester, N., y Shannon, C. (1955). *A proposal for the Darmouth Summer Research Project on Artificial Intelligence*. <http://jmc.stanford.edu/articles/dartmouth/dartmouth.pdf>

- Medianovskyi, K. y Pietarinen, A.-V. (2022). On Explainable AI and Abductive Inference. *Philosophies*, Vol. 7, No. 35. <https://doi.org/10.3390/philosophies7020035>
- Miranda-Rojas, R. (2018). “Intuición, racionalidad y confiabilidad”, *Cinta Moebio*, Vol. 62, pp. 261-273. <http://dx.doi.org/10.4067/S0717-554X2018000200261>
- Leib, R. (2021). “Interaction between Language and the other Symbolic Forms”, en: Simon Truwant, ed., *Interpreting Cassirer. Critical Essays*, Cambridge: Cambridge University Press, pp. 11-13
- OpenAI. (2019). *Emergent Tool from Multi-Agent Interaction*. Disponible Online: <https://openai.com/blog/emergent-tool-use/>
- Park, W. (2022). “How to Make AlphaGo’s Children Explainable”. *Philosophies*, Vol. 7, No. 55. <https://doi.org/10.3390/philosophies7030055>
- Peirce, Ch. (1998). “Pragmatism as the Logic of Abduction (Lecture VII)”, en: Peirce Edition Project, eds., *The Essential Peirce. Selected Philosophical Writings. Vol. 2 (1893-1913)*, Bloomington and Indiana: Indiana University Press, pp. 226-240.
- Platón, (2015). *Diálogos*, Traducción y notas de Ma. Santa Cruz; A. Vallejo Campos; N. Luis Cordero, Madrid: Gredos, 2015.
- Silver, D., Schrittwieser, J., Simonyan, K., *et al.* (2017). “Mastering the game of Go without human knowledge”, *Nature*, Vol. 550, pp. 354–359. <https://doi.org/10.1038/nature24270>