UNIVERSIDAD DE GRANADA

Historia de las Matemáticas

Problemas éticos y morales de la Inteligencia Artificial

Evolución a lo largo de la Historia

Fernando de la Hoz Moreno
Laura Gómez Garrido
Antonio Martín Ruiz

Curso 2019-2020

1 de diciembre de 2019

Índice

1. Introducción **2**

2. ¿Qué es la Inteligencia Artificial? **2**
2.1. Comportamiento humano: el enfoque de la prueba de Turing 2
2.2. Pensar como un humano: el enfoque del modelo cognitivo 3
2.3. Pensamiento racional: el enfoque de las leyes del pensamiento 3
2.4. Actuar de forma racional: el enfoque del agente racional 4

3. Historia de la Inteligencia Artificial **4**
3.1. Génesis de la Inteligencia Artificial(1943-1955) 4
3.2. Nacimiento de la Inteligencia Artificial(1956) 5
3.3. Entusiasmo inicial, grandes esperanzas (1952-1969) 5
3.4. Una dosis de realidad (1966-1973) 6
3.5. Sistemas basados en conocimiento: ¿clave del poder? 7
3.6. La IA se convierte en una industria 8
3.7. Regreso de las redes neuronales 8
3.8. La IA se convierte en una ciencia 8
3.9. Estado del arte 9

4. Problemas éticos **9**
4.1. Objecciones de Turing 9
4.1.1. La objeción teológica 10
4.1.2. La objeción de la *cabeza en la arena* 10
4.1.3. La objección matemática 10
4.1.4. El argumento de la conciencia 11
4.1.5. Argumentos sobre diversas incapacidades 11
4.1.6. La objeción de Lady Lovelace 12
4.1.7. El argumento de la continuidad del sistema nervioso 12
4.1.8. El argumento de la información del comportamiento 12
4.1.9. El argumento de la percepción extrasensorial 13
4.2. Seguridad en IA 13
4.2.1. Evitar efectos secundarios negativos 13
4.2.2. Evitar saboteo del sistema de recompensas 15
4.2.3. Exploración segura 15
4.3. Uso malicioso de la IA 17

5. Conclusiones **18**

1. Introducción

2. ¿Qué es la Inteligencia Artificial?

Existen diferentes visiones sobre qué es la inteligencia artificial. Existe una división entre las que se refieren a proceso mental o razonamiento y las que se refieren a conducta. También se pueden dividir entre las que miden el éxito en función de la fidelidad en la forma de actuar de los humanos, o a un concepto ideal de inteligencia (racionalidad). Diferenciamos por tanto cuatro enfoques:

- Sistemas que piensan como humanos
- Sistemas que actúan como humanos
- Sistemas que piensan racionalmente
- Sistemas que actúan racionalmente

Veamos cada uno de estos enfoques.

2.1. Comportamiento humano: el enfoque de la prueba de Turing

La prueba de Turing, propuesta por Alan Turing (1950) se diseñó para proporcionar una definición operacional y satisfactoria de la inteligencia. Consiste en una prueba basada en la incapacidad de diferenciar entre entidades inteligentes y seres humanos. El computador supera la prueba si un evaluado humano no es capaz de distinguir si las respuestas a una serie de preguntas planteadas son de una persona o no. Para ello un computador debe poseer las siguientes capacidades:

- Procesamiento del lenguaje natural para poder comunicarse
- Representación del conocimiento para almacenar lo que conoce o siente
- Razonamiento automático para utilizar la información almacenada para responder preguntas y extraer conclusiones
- Aprendizaje automático para adaptarse a nuevas circunstancias y detectar y extrapolar patrones.

La prueba de Turing evita deliberadamente la interacción física directa entre evaluador y computador, ya que se centra únicamente en la inteligencia. Existe una Prueba Global de Turing que incluye una señal de vídeo para que evaluador valore la capacidad de percepción del evaluado, y la capacidad para el evaluador de pasar objetos físicos. Para superar la prueba

global el computador debe estar también dotado de

- Visión computacional para percibir objetos
- Robótica para manipular objetos

La prueba sigue vigente, pero los investigadores de la IA han dedicado poco esfuerzo a la evaluación de sus sistemas con la Prueba de Turing, por creer que es más importante el estudio de los principios en los que se basa la inteligencia que duplicar un ejemplar de la misma.

2.2. Pensar como un humano: el enfoque del modelo cognitivo

Para poder decir que un programa dado piensa como un humano es necesario contar con un mecanismo para determinar cómo piensan los humanos. Una vez se cuente con una teoría los suficientemente precisa sobre cómo trabaja la mente, se podrá expresar esa teoría en la forma de un programa de computador. En el campo interdisciplinario de la ciencia cognitiva convergen modelos computacionales de IA y técnicas experimentales de psicología intentado elaborar teorías precisas y verificables sobre el funcionamiento de la mente humana.

En los comienzos de la IA existía una confusión entre las distintas aproximaciones: se argumentaba que un algoritmo resolvía adecuadamente una tarea y que por tanto era un buen modelo de representación humana, o viceversa. La correcta diferenciación ha permitido que tanto IA como ciencia cognitiva se desarrollen más rápidamente. Los dos campos se alimentan entre sí, especialmente en las áreas de visión y lenguaje natural.

2.3. Pensamiento racional: el enfoque de las leyes del pensamiento

Los silogismos de Sócrates son el primer intento de codificar la "manera correcta de pensar", es decir, un proceso de razonamiento irrefutable. Son el inicio del campo que hoy día conocemos como lógica.

En el siglo XIX se desarrolla una notación precisa para definir sentencias sobre elementos del mundo y especificar relaciones entre ellos. La tradición logista dentro del campo de la inteligencia artificial trata de construir sistemas inteligentes a partir de la lógica.

Este enfoque presenta dos obstáculos. No es fácil transformar el conocimiento informar y expresarlo en términos formales que requieren de notación lógica, particularmente cuando el conocimiento que se tiene sobre algo no es total. En segundo lugar, hay una diferencia entre poder resolver el problema y poder hacerlo en la práctica. Incluso problemas con apenas una docena de datos pueden agotar los recursos computacionales de cualquier computador. Estos obstáculos están presentes en todo intento de construir sistemas de razonamiento computacional pero surgieron por primera vez en la tradición lógica.

2.4. Actuar de forma racional: el enfoque del agente racional

Un agente es algo que razona, pero los agentes informáticos deben tener otros atributos como estar dotados de controles autónomos, percibir su entorno, persistir un periodo de tiempo prolongado, adaptarse a cambios y ser capaces de alcanzar objetivos diferentes. Un agente racional es aque que actúa con la intención de alcanzar el mejor resultado o, cuando hay incertidumbre, el mejor resultado esperado.

El agente racional no solo se centra en hacer inferencias correctas, sino que además debe llevar a cabo las acciones. Efectuar una inferencia correcta no depende de la racionalidad, ya que hay situaciones en las que no existe una acción correcta y hay que tomar decisiones. Existen formas de actuar racionalmente que no implican inferencia. La racionalidad perfecta (hacer siempre lo correcto) no es siempre posible en entornos complejos. La demanda computacional que esto implica es demasiado grande. Es por ello que podemos contar con que los agentes racionales tienen una racionalidad limitada (actuar adecuadamente cuando no se cuenta con el tiempo suficiente para efectuar todos los cálculos deseables).

3. Historia de la Inteligencia Artificial

3.1. Génesis de la Inteligencia Artificial(1943-1955)

Warren McCulloch y Walter Pitts (1943) han sido reconocidos como los autores del primer trabajo de IA, partiendo tres fuentes: conocimientos sobre la fisiología básica y funcionamiento de las neuronas en el cerebro, el análisis formal de la lógica proposicional de Russel y Whitehead y la teoría de computación de Turing. Propusieron un modelo constituido por neuronas artificiales y mostraron, por ejemplo, que cualquier función de cómputo podía calcularse mediante alguna red de neuronas interconectadas y que todos los conectores lógicos (and, or, not, etc) se podrían implementar utilizando estructuras de red sencillas. También sugirieron el aprendizaje por parte de unas neuronas adecuadamente definidas.

Donald Hebb (1949) propuso y demostró una sencilla regla de actualización para modificar las intensidades de las conexiones entre neuronas. Actualmente es denominada *Regla de Aprendizaje Hebbiano o de Hebb* y sigue vigente en la actualidad.

Alan Turing articuló primero una visión de la IA en su artículo *Computing Machinery and Intelligence* en 1950 donde introdujo la prueba de Turing, el aprendizaje automático, los algoritmos genéricos y el aprendizaje por refuerzo.

3.2. Nacimiento de la Inteligencia Artificial(1956)

Fue en el verano de 1956 cuando se decidió el nombre que esta nueva área de la investigación adoptaría: *Inteligencia Artificial*. Esto sucedió durante un taller en Darmouth con duración de dos meses, organizado por John McCarthy, Minsky, Claude Shannon y Nathaniel Rochester que tuvo un total de 10 asistentes entre los cuales se incluían Trenchad More de Princeton, Arthur Samuel de IBM, y Ray Solomoff y Oliver Selfridge del MIT.

Aunque no se hicieron grandes progresos en la investigación conjunta, destacaron particularmente dos investigadores del Carnegie Tech (Universidad de Carnegie Mellon), Allen Newel y Helbert Simon quienes contaban con un programa de razonamiento, el *Teórico Lógico(TL)*. Al término del taller, dicho programa ya era capaz de demostrar gran parte de los teoremas del Capítulo 2 de *Principia Matemática* de Russel y Whitehead, incluso con demostraciones más cortas que las aportadas por los propios autores.

Newell y Simon también desarrollaron un lenguaje de procesamiento de listas, IPL, para poder escribir el TL. No disponían de un compilador y lo tradujeron a código máquina a mano. Para evitar errores, trabajaron en paralelo, diciendo en voz alta números binarios, conforme escribían cada instrucción para asegurarse de que ambos coincidían.

3.3. Entusiasmo inicial, grandes esperanzas (1952-1969)

El temprano éxito de Newell y Simon siguió el del sistema general de problemas, o SRGP, el cual, a diferencia del Teórico Lógico, desde un principio de se diseñó para que imitara protocolos de resolución de problemas de los seres humanos. Siendo probablemente, SRGP el primer programa que incorporó el enfoque de "pensar como un humano". Este y otros éxitos llevaron a Newel y Simon a a formular en 1976 la famosa hipótesis del *sistema de símbolos físicos* que afirma que

> un sistema de símbolos físcos tiene los medios suficientes y necesarios para generar una acción inteligente

, es decir, que cualquier sistema (humano o máquina) que exhibiese inteligencia debería operar manipulando estructuras de datos compuestas por símbolos.

Arthur Samuel (1952) escribió una serie de programas para el juego de damas que eventualmente aprendieron a jugar hasta el nivel de un amateur, echando por tierra la idea de que los computadores sólo pueden hacer lo que se les dice: su programa aprendió a jugar mejor que su creador.

Herter Gelernter (1959) construyó el demostrador de teoremas de geometría (DTG), capaz de demostrar teoremas que muchos estudiantes de matemáticas podían encontrar muy complejos de resolver.

En 1958, John McCarthy realizó tres grandes contribuciones:

- Definió el lenguaje de alto nivel Lisp, el segundo lenguaje de programación de alto nivel más antiguo que aún hoy se utiliza, al ser creado un año después que FORTRAN, y que se convertiría en el lenguaje de programación dominante de la IA.
- Publicó el artículo *Programs with Common Sense*, en el que describía el Generador de Consejos. Este era un programa hipotético que podría considerarse el primer sistema de IA completo y que, a diferencia del TL y DTG, utilizaba conocimiento general del mundo. Era capaz de aceptar nuevos axiomas durante el curso normal de operación, permitiéndole así ser competente en nuevas áreas sin necesidad de reprogramación.
- Junto a otros compañeros, inventó el tiempo compartido para tratar de solucionar el acceso a los escasos y costosos recursos de cómputo.

Minsky supervisó el trabajo de una serie de estudiantes del MIT que eligieron un número de problemas limitados cuya solución pareció requerir inteligencia, los cuales se conocen como *micromundos*. El micromundo más famoso fue el mundo de bloques, que consiste en un mundo de bloques sólidos colocados sobre una mesa y cuya tarea típica es la reordenación de los bloques utilizando la mano de un robot que es capaz de tomar un sólo bloque cada vez.

El trabajo realizado por McCulloch y Pitts con redes neuronales hizo florecer esta área. Winograd y Cowan mostraron en 1963 cómo un gran número de elementos podrían representar un concepto individual de forma colectivo, llevando consigo un aumento proporcional en robustez y paralelismo. ****(FIXME: Hay más detalles, pero en caso de ponerlos habría que buscar sus definiciones: adalines y perceptrones)****

3.4. Una dosis de realidad (1966-1973)

El primer tipo de problemas surgió porque la mayoría de los primeros programas contaban con poco o ningún conocimiento de las materia objeto de estudio; obtenían resultados gracias a sencillas manipulaciones sintácticas. Una anécdota divertida surge con los problemas que surgieron al tratar de traducir algunos textos entre diferentes lenguajes de forma automática.

El segundo problema fue que muchos de los problemas que se estaban tratando de resolver mediante la IA eran intratables. El optimismo que acompañó el logro de la demostración de teoremas pronto se vio eclipsado cuando los investigadores fracasaron en la demostración de teoremas más de unas pocas decenas de condiciones.

> El hecho de que, en principio, un programa sea capaz de encontrar una solución no implica que tal programa encierre todos los mecanismos necesarios para encontrar la solución en la práctica.

El tercer obstáculo se derivó de las limitaciones inherentes a las estructuras básicas que se

utilizaban en la generación de la conducta inteligente. Por ejemplo, en 1969, en el libro de Minsky y Papert, *Perceptrons*, se demostró que si bien era posible lograr que los perceptrones (una red neuronal simple) aprendieran cualquier cosa que pudiesen representar, su capacidad de representación era muy limitada.

3.5. Sistemas basados en conocimiento: ¿clave del poder?

La resolución de problemas durante la primera década de investigación de la IA se centraba en los mecanismos de búsqueda de propósito general (métodos débiles). La alternativa a los métodos débiles es el uso de conocimiento específico del dominio. Para ello es necesario saber de antemano la correspondiente respuesta al problema.

Uno de los primeros ejemplos fue DENDRAL, programa diseñado para inferir la estructura molecular, a partir de la fórmula molecular y la información del espectrómetro de masas. La primera versión de DENDRAL utilizaba la fuerza bruta para generar todas las posibilidades y ver cual coincidía con la información del espectrómetro de masas, pero era inviable. Consultaron a químicos analíticos y vieron que ellos buscaban patrones de picos conocidos en el espectrómetro para reconocer subestructuras y reducir el número de posibilidades. DENDRAL fue el primer sistema de conocimiento intenso con éxito.

Teniendo en cuenta esta lección investigadores de Standford dieron comienzo al Proyecto de Programación Heurística, PPH, dedicados a determinar como aplicar los sistemas expertos en áreas de la actividad humana.

El siguiente gran esfuerzo se realizó en el área de diagnóstico médico. Diseñaron el programa MYCIN para diagnosticar infecciones sanguíneas. Sus diagnósticos eran tan buenos como los de un experto. Contaba con 450 reglas.

Se distinguía de DENDRAL en dos aspectos principalmente. MYCIN no poseía un modelo teórico desde el cual obtener las reglas, como DENTRAL. Tuvieron que obtenerse a través de entrevistas con expertos. En segundo lugar MYCIN debía reflejar la incertidumbre inherente al conocimiento médico.

También se utilizó el conocimiento del dominio en el área de la comprensión del lenguaje natural, pero no se pudo resolver algunos de los problemas que ya habían aparecido con la traducción automática ocasionados por el análisis sintáctico.

El crecimiento de aplicaciones para solucionar problemas del mundo real provocó el aumento de la demanda de esquemas de representación del conocimiento, como Prolog basado en lógica u otros basados en la noción de marcos de Minsky, mas estructurado y jerárquico.

3.6. La IA se convierte en una industria

En la década de los 80 se comercializaban cientos de sistemas expertos a empresas las cuales conseguían ahorrarse millones de dolares con estos. EEUU, Japón y Reino Unido hicieron grandes inversiones en IA. La industria de la IA paso de unos pocos de millones de dolares en 1980 a billones de dolares en 1988. Poco después llegó .el invierno de la IA"que afectó a muchas empresas que no fueron capaces de desarrollar extravagantes productos prometidos.

3.7. Regreso de las redes neuronales

La informática abandonó el campo de las redes neuronales a finales de los 70, pero continuó en otros campos como la física o la psicología. El impulso mas fuerte se produjo a mediados de los 80, cuando por lo menos cuatro grupos distintos reinventaron el algoritmo de aprendizaje de retroalimentación. Este se aplicó a problemas de aprendizaje de la informática y la psicología, y la gran difusión que conocieron los resultados suscito un gran entusiasmo.

Aquellos modelos de IA, llamados conexionistas, fueron vistos por algunos como competidores tanto de los modelos simbólicos, como de la aproximación lógica. La tendencia actual es que la aproximación conexionista y simbólica son complementarias.

3.8. La IA se convierte en una ciencia

En los últimos años se ha producido una revolución tanto en el contenido como en la metodología de trabajo en la IA. Actualmente es mas usual el desarrollo sobre teoría ya existentes que proponer nuevas, tomar como base teoremas y evidencias experimentales más que intuición. La IA se fundó en parte en el marco de una rebelión contra las limitaciones de los campos existentes como la teoría de control o la estadística, y ahora abarca estos campos.

En términos metodológicos, se puede decir, ya forma parte del ámbito de los métodos científicos. Para que se acepten, las hipótesis deben someterse a rigurosos experimentos empíricos, y los resultados deben analizarse estadísticamente para identificar su relevancia.

En años recientes, los modelos de Markov ocultos, han dominado el área de reconocimiento del habla gracias a apoyarse una rigurosa teoría matemática que ha permidito basar las investigaciones en resultados obtenidos durante décadas y en generar los modelos a partir de grandes volúmenes de datos mediante un proceso de aprendizaje.

La utilización de metodologías mejoradas y marcos teóricos en las redes neuronales, ha permitido que alcance un grado de conocimiento comparado con otras técnicas similares en estadística, reconocimiento de patrones y aprendizaje automático. Como resultado de estos desarrollos, la tecnología minería de datos ha generado una gran industria.

A finales de los 80 el formalismo de las redes de Bayes apareció para facilitar la representación eficiente y el razonamiento riguroso en las situaciones con conocimiento incierto. Este enfoque supera muchos problemas de los sistemas de razonamiento probabilístico de los 60 y 70 y ahora domina la investigación de la IA en razonamiento incierto y los sistemas expertos. Esta aproximación facilita el aprendizaje a partir de la experiencia combinando lo mejor de la IA clásica y las redes neuronales.

3.9. Estado del arte

La inteligencia artificial está obteniendo grandes resultados en la automatización de una gran variedad de tareas. Veamos algunas de las aplicaciones actuales.

En el reconocimiento de imágenes se están obteniendo resultados mejores que los obtenidos por humanos. En la generación de imágenes son capaces de generar imágenes sintéticas prácticamente indistinguibles de fotografías.

<!— TODO: añadir gráficas pag 14/15 de Malicious Use—>

Los sistemas de inteligencia artificial están obteniendo asombrosos resultados en gran variedad de juegos competitivos, desde el ajedrez al Go y en e-Sports como Dota 2, gracias a técnicas que buscan de manera creativa estrategias exitosas en el largo plazo, apoyándose en objetivos auxiliares y aprendiendo de ejemplos humanos.

Otros campos en los que se empiezan a obtener resultados son el reconocimiento de voz, la comprensión del lenguaje y la navegación automática de vehículos.

<!— TODO: probablemente haya que completarlo—>

4. Problemas éticos

4.1. Objecciones de Turing

En 1950 fue publicado, en la revista Mind, el artículo *Çomputing Machinery and Intelligence"* escrito por Alan Turing, el cual se centraba en tratar el tema de la Inteligencia Artificial.

El artículo comienza dando en consideración la pregunta de *"¿Pueden pensar las máquinas?"*, dejando en claro que la principal intención del artículo es discutir esta cuestión. Así es como el documento destaca por dos grandes cosas: La primera mención al público del conocido *Test de Turing* y el planteamiento y respuesta de algunas de las *Objeciones de Turing*. En este apartado nos centraremos en estas últimas.

Estas objeciones, son los hipotéticos argumentos que Turing se imaginaba que recibiría como respuesta negativa a la pregunta formulada anteriormente. De esta forma, el autor se dedicó a desmentir cada una de estas objeciones en el mismo artículo. A continuación, comentaremos cada una de estas objeciones:

4.1.1. La objeción teológica

> El pensamiento es una función del alma inmortal del hombre. Dios ha proporcionado un alma inmortal a todos los hombres y mujeres, pero no así a ningún otro animal, ni tampoco a las máquinas. Por consiguiente, ningún animal o máquina puede pensar.

Aquí deja bastante claro tanto al inicio como al final de su razonamiento lo poco de acuerdo que está con esta opinión religiosa y, pese a ello, trata de rebatirla utilizando argumentos de índole teológica.

Su contraargumento se basa la propia suposición que la cita hace sobre el alcance de los poderes de Dios.

Concretamente, la cita afirma que Dios sólo puede conferir de alma inmortal a los hombres y mujeres, cuando Él tiene la libertad de conferir alma a un animal si así lo desea. De la misma forma, si así lo quisiera, Dios podría conferirle a una máquina de un alma inmortal, si así lo deseara.

Así, al crear máquinas pensantes, seríamos instrumentos de Su voluntad para crear recintos para las almas que él crea, de la misma forma que si procreásemos hijos.

4.1.2. La objeción de la *cabeza en la arena*

> Las consecuencias de que las máquinas pensaran serían demasiado terribles. Esperemos y creamos que no pueden hacerlo.

Aquí comenta que nos gusta pensar que el hombre es, en cierto modo, superior al resto de los seres. Concretamente, nos gustaría poder demostrar que es necesariamente superior, puesto que así no habría peligro de que perdiera su posición dominante. Añade, que no tiene sentido tratar de seguir refutando este argumento y, finaliza bromeando con que quizás quienes sean afines con este pensamiento deberían buscar consuelo en las transmigración de las almas.

4.1.3. La objección matemática

> Existen muchos resultados de lógica matemática que pueden utilizarse para demostrar que hay limitaciones al potencial de las máquinas de estado discreto. [...] Este es el resultado

> matemático: se afirma que prueba que las máquinas adolecen de una incapacidad a la que no se encuentra sujeto el intelecto humano.

La respuesta breve que ofrece Turing a esta afirmación, no es ni más ni menos que durante el propio razonamiento se asume que las limitaciones que les ponen a las máquinas no se aplican a los seres humanos, cuando esto no es necesariamente cierto.

Después de todo, los seres humanos respondemos erróneamente con frecuencia y estos errores no nos hace ser menos humanos.

Entonces, ¿por qué si una máquina comete errores que también son cometidos por humanos, ya afirmamos haber demostrado que no es pensante? Es dicha pregunta la que utiliza Turing para rebatir el anterior argumento "matemático".

4.1.4. El argumento de la conciencia

> No podremos aceptar que la máquina iguala al cerebro hasta que una máquina pueda escribir un soneto o componer un concierto en respuesta a pensamientos y emociones experimentadas y no mediante una cascada aleatoria de símbolos. (Esto es, no sólo escribir el soneto, sino saber que ha sido escrito.) Ningún mecanismo podría sentir placer por sus éxitos (y no meramente emitir artificialmente una señal, fácil artilugio), experimentar pesar cuando se funden sus válvulas, ni sentirse enternecido por los halagos o miserable por sus errores, ni encantada por el sexo o enfadada o deprimida cuando no consigue lo que desea.
> — Jefferson, 1949

Aquí, Turing comenta que la única forma de estar seguros de si una máquina piensa o no, es de hecho, *ser* la máquina y sentirse uno mismo pensar; puesto que así podríamos describir estos sentimientos al mundo pero nadie se sentiría justificado por prestar atención. De la misma forma, para saber si un hombre concreto piensa, tendríamos que ser ese hombre en particular.

De esta forma, dos individuos pueden razonar que sólo ellos piensan y el contrario no. Pero pese a ello, en lugar de entrar en bucle infinito sobre cuál de los dos piensa, se tiende a recurrir al convenio de que ambos piensan.

4.1.5. Argumentos sobre diversas incapacidades

Estos argumentos suelen ser de la forma: *.Acepto que puedas hacer que las máquinas hagan todo lo que hasta ahora has mencionado, pero nunca podrás hacer que una de ellas haga X"*.

Aquí, tras dar varios ejemplos sobre cosas que fueron dichas que las máquinas no podrían hacer y que a largo plazo sí podría ser implementado, comenta que decir que una máquina no tiene diversidad de conductas se traduce en decir que no tiene gran capacidad de almacenamiento. Conforme pasa el tiempo, cada vez podemos ver que la falta de almacenamiento no es

precisamente un problema del que suelan disponer las máquinas.

Además, compara este argumento con el argumento de Jefferson y lo ve como una forma de disfrazar el argumento de la conciencia.

4.1.6. La objeción de Lady Lovelace

> La máquina no pretende crear nada. Puede hacer lo que sea que sepamos ordenarle. - Ada Lovelace, 1842

Aquí, Turing comenta que la falta de conocimiento de Lovelace respecto al conocimiento matemático y computacional del tiempo de Turing fue lo que, dentro del contexto de Ada, la llevase a realizar dicha afirmación.

Además, nos remite a la sección *"Máquinas que aprenden"* del mismo artículo, donde aborda en mayor profundidad y detalle esta temática.

4.1.7. El argumento de la continuidad del sistema nervioso

Turing admite que el sistema nervioso no es una máquina de estado discreto. Un pequeño error en la información acerca de las dimensiones del impulso nervioso que incide en una neurona puede marcar una gran diferencia en las dimensiones del impulso de salida. Podría argüirse que, siendo así, no podemos esperar ser capaces de imitar el comportamiento del sistema nervioso con un sistema de estado discreto.

Es cierto que una máquina de estado discreto debe ser diferente de una máquina continua. Sin embargo, si nos apegamos a las condiciones del Test de Turing, el examinador no tendría ninguna ventaja con esta diferencia.

4.1.8. El argumento de la información del comportamiento

> No es posible producir un conjunto de reglas que pretenda describir lo que una persona debe hacer en cada grupo de circunstancias concebible. Podría, por ejemplo, haber una regla que dictara que debemos detenernos al ver la luz roja de un semáforo y avanzar cuando la luz cambie a verde. Entonces, ¿qué sucedería si por algún desperfecto ambas aparecieran al mismo tiempo? Tal vez se decidiría que lo más seguro sería detenerse. No obstante, más adelante podría surgir otra dificultad a raíz de esta decisión. Intentar proporcionar reglas de conducta que cubran cualquier eventualidad, incluso las que surjan a partir de las luces de los semáforos, parecería imposible. Concuerdo con todo esto

Si este argumento fuera cierto, entonces deberíamos de ser capaces de descubrir por observación lo suficiente acerca de ella para predecir su comportamiento futuro en un tiempo

razonable. Turing reta al lector a ser capaz de conseguir estas leyes de comportamiento para una pequeña máquina que implementó en Manchester.

4.1.9. El argumento de la percepción extrasensorial

Turing no tiene un argumento viable en contra de estas afirmaciones, en caso de considerarlas ciertas. Además, comenta que en caso de considerarlas ciertas, ya no podríamos considerar que nuestros cuerpos se mueven de acuerdo a las leyes conocidas de la física, ni con algunas otras aún no descubiertas.

4.2. Seguridad en IA

Un accidente es una situación en la cual el diseñador humano tenía en mente cierto objetivo o tarea pero el sistema diseñado para dicha tarea produce resultados inesperados o dañinos. Podemos clasificar problemas de seguridad según en qué momento del proceso se producen errores.

Primeramente, el diseñador especifica de manera incorrecta la función objetivo, la cual produce resultados dañinos. Si se establece como único objetivo una tarea específica ignorando otros aspectos del ambiente que pueden ser dañinos si son afectados pueden generarse efectos secundarios negativos. Una definición incorrecta del objetivo puede también generar un saboteo del sistema de recompensas (*reward hacking*) de manera que formalmente se maximice el objetivo indicado pero esto no produzca el efecto deseado.

El diseñador conoce la función objetivo correcta pero la manera de evaluarla es demasiado costosa. Malas extrapolaciones de muestras limitadas pueden generar comportamientos dañinos.

Por último, el diseñador puede haber especificado la función objetivo correcta de tal manera que se obtiene un comportamiento correcto, pero se producen malas decisiones debido a unos datos de entrenamiento pobres o a un modelo insuficientemente expresivo. La exploración de agentes puede causar consecuencias negativas y los algoritmos de aprendizaje automático no interpretables pueden tomar malas decisiones con entradas muy diferentes a aquellas con las que han sido entrenados.

Tenemos por tanto cinco causas de accidentes que deben ser contempladas.

4.2.1. Evitar efectos secundarios negativos

Supongamos un agente cuyo objetivo es lograr mover una caja de un lado a otro de la habitación. A veces la manera más efectiva de lograr el objetivo implica hacer algo no

relacionado y destructivo para el resto del medio ambiente, como en este caso podría ser derribar un jarrón que se encuentra por el camino.

Podríamos diseñar el agente para darle una recompensa negativa por tocar el jarrón y habríamos terminado. Pero que pasa si hay muchas cosas dirruptivas que el agente podría hacer al medio como como cortocircuitar una toma de corriente o dañar las paredes de una habitación. Eso puede no ser factible identificar y penalizar cada posible irrupción.

En términos generales, para un agente que opera en un entorno grande y multifacético, una función objetivo que se centra en un solo aspecto del entorno puede expresar implícitamente indiferencia sobre otro aspecto del medio ambiente. Un agente que optimice esta función objetivo podría producir grandes disrupciones en el entorno si hacerlo le proporciona una pequeña ventaja para la tarea en cuestión.

Al igual que las funciones objetivo mal especificadas, los efectos secundarios asociados a cada tarea individual, podrían ser responsabilidad del diseñador para incluir como parte del diseño de la función objetivo correcta. Sin embargo, los efectos pueden ser muy similares incluso para tareas muy diversas, como por ejemplo derribar muebles, por lo que vale la pena atacar el problema en general. Un enfoque exitoso podría ser transferible entre tareas, contrarrestando uno de los mecanismos por los que se producen funciones objetivos incorrectas. Algunos enfoques para combatir estos problemas son:

- Definir un regularizador de impacto: si no queremos efectos secundarios, lo natural parece penalizar los cambios en el entorno. Esto se haría dándole preferencia por las formas de lograr su objetivo con efectos secundarios mínimos o dándole al agente un "presupuesto de impacto limitado".
- Aprender un regularizador de impacto: un enfoque alternativo y más flexible, en lugar de definir un un regularizador de impacto a través de la capacitación en muchas tareas. Esto sería una instancia de transferencia de aprendizaje.
- Penalizar la influencia: además de no hacer cosas que tengan efectos secundarios, también podríamos hacer que el agente prefiera no colocarse en posiciones donde pueda hacer cosas que tienen efectos secundarios, aunque pudieran ser convenientes.
- Enfoques de múltiples agentes: evitar los efectos secundarios se realiza con la intención de evita las externalidades negativas. Si a todos les gusta un efecto secundario, no hay necesidad de evitarlo. Lo que de verdad nos gustaría hacer es comprender a todos los demás agentes (incluido los humanos) y asegurarnos de que nuestras acciones no perjudiquen sus intereses.
- Incertidumbre de recompensa: queremos evitar efectos secundarios imprevistos porque el medio ambiente ya es bastante bueno según nuestras preferencias. En lugar de darle a un agente una función de recompensa única, podría ser incierto sobre la función de recompensa, con una distribución de probabilidad previa que refleja la propiedad de que

los cambios aleatorios tienen más probabilidad de ser malos que buenos.

4.2.2. Evitar saboteo del sistema de recompensas

El saboteo del sistema de recompensas puede llevar a comportamientos inesperados que son soluciones formales del problema, que podrían ser potencialmente dañinos. Varias formas de este fenómeno han sido investigadas desde una perspectiva teórica:

- Metas parcialmente observadas
- Sistemas complicados
- Recomensas abstractas
- Ley de Goodhart
- Bucles de retroalimentación
- Incrustación ambiental

4.2.3. Exploración segura

Todos los agentes de aprendizaje autónomos necesitan participar en la exploración, tomar acciones que no parecen ideales dada la información actual, pero ayudan al agente a conocer su entorno actual. Sin embargo, la exploración puede ser peligrosa, ya que implica tomar acciones cuyas consecuencias el agente puede no entender bien. En entornos de juguete, como en videojuegos, las consecuencias no tienen gran gravedad. Pero el mundo real puede ser menos indulgente. Las acciones mal elegidas pueden destruir al agente o atraparlo en estados de los que no puede salir, como por ejemplo los helicópteros robot pueden caer al suelo o dañar la propiedad. Políticas de exploración comunes eligen una acción al azar o ven acciones inexploradas optimistas y por tanto no intentan evitar estas situaciones optimistas. Estrategias de exploración mas sofisticadas adoptan una política de exploración coherente a escalas temporales más prolongadas, lo que en realidad podria tener un mayor potencial de daño, ya que una mala política elegida coherentemente puede ser mas perjudicial que simples acciones aleatorias. Sin embargo, intuitivamente parece que a menudo debería ser posible para predecir que acciones son peligrosas y explorar de una manera que las evite, incluso cuando no tenemos mucha información sobre el medio ambiente.

En la práctica, se pueden evitar estos problemas simplemente codificando evitar los comportamientos catastróficos. Por ejemplo un robot helicóptero podría programarse para anular su política con una secuencia codificada para evitar colisiones (como girar las hélices para gar altitud) siempre que este demasiado cerca del suelo. Este enfoque funciona bien cuando solo hay algunas cosas que podrían salir mal, y los diseñadores las conocen todas con anticipación. Pero

como agentes, volverse mas autónomos y actuar en dominios más complejos, puede ser cada vez mas difícil anticipar cualquier posible falla catastrófica como un agente ejecutando una red eléctrica o una operación de búsqueda y rescate. La codificación dura contra todo lo posible hace poco probable que el fracaso sea factible en estos casos, por lo que un enfoque basado en principios para prevenir en la exploración perjudicial parece esencial. Incluso en casos como el helicóptero robot, un enfoque basado en principios simplificaría el diseño del sistema y reduciría la necesidad de ingeniería específica del dominio.

Hay una literatura considerable sobre exploración segura, pero aquí describiremos rutas generales que ha tomado esta investigación.

- Criterios de rendimiento sensibles al riesgo: un conjunto de literatura existente considera cambiar los criterios de optimización de la recompensa total esperada a otros objetivos que son mejores en prevención de eventos raros y catastróficos. Estos enfoques implican optimizar el rendimiento en el peor de los casos, o garantizar que la probabilidad de un muy mal desempeño sea pequeña o penaliza la variación en el desempeño.

- Demostraciones de uso: la exploración es necesaria para que el agente encuentre los estados que son necesarios para un rendimiento casi óptimo. Podemos evitar la necesidad de explorar en conjunto si en su lugar utilizamos RL inverso o aprendizaje de aprendizaje, donde el algoritmo de aprendizaje esta provisto de trayectorias expertas de comportamiento casi óptimo. Recientes progresos en el aprendizaje por refuerzo inverso usando redes neuronales profundas para aprender la función de coste o política sugiere que podría ser posible reducir la necesidad de exploración en sistemas avanzados RL por entrenamiento en un conjunto pequeño de demostraciones. Tales demostraciones podrían ser usadas para crear una política de referencia.

- Exploración simulada: cuanto más podamos hacer nuestra exploración en entornos simulados, en lugar de en el mundo real, hay menos oportunidades de catástrofe. Probablemente siempre será necesario hacer exploraciones en el mundo real, ya que muchas situaciones complejas no pueden ser capturadas por un simulador, pero podría ser posible aprender sobre el peligro en simulación y luego adoptar una política mas conservadora .[ex]ploración seguraçuando actúe en el mundo real. El entrenamiento de agentes RL en entornos simulados son ya muy comunes, por lo que los avances en la "simulación centrada en la exploración"podrían incorporarse fácilmente a los flujos de trabajo actuales. En los sistemas qe implican un ciclo continuo de aprendizaje y despliegue, pueden ser interesantes problemas de investigación asociados con la forma segura de actuar de forma incremental las políticas dadas basadas en simulaciones.

- Exploración limitada: si sabemos que cierta porción del espacio es segura y que incluso la peor acción dentro de ella puede recuperarse o limitarse en daños, podemos permitir al agente que explore libremente dentro de esos limites. Si tenemos un modelo, podemos extrapolar un paso hacia delante y preguntar si esa acción nos va a llevar fuera del espacio de estados seguros. La seguridad puede definirse como permanecer en una región

ergódica del espacio de estados tal que las acciones son reversibles, o como limitación de la probabilidad de una gran recompensa negativa para pequeños valores. Otros enfoques usan funciones de seguridad y rendimiento separadas e intentan obedecer las restricciones sobre la función de seguridad con alta probabilidad.

- Supervisión de políticas de confianza: si tenemos una política de confianza y un modelo de entorno, puede limitar la exploración a acciones de las que la política de confianza cree que podemos recuperarnos. Esta bien bucear hacia el suelo, siempre que sepamos que podemos salir de la inmersión a tiempo.
- Supervisión humana: otra posibilidad es verificar las acciones potencialmente inseguras con un humano. Desafortunadamente, este problema se encuentra con el problema de supervisión escalable: el agente puede necesitar realizar demasiadas acciones exploratorias para que la supervisión humana sea práctica, o tal vez sea necesaria demasiada inmediatez para que los humanos los juzguen. Un desafío clave para hacer que esto funcione es hacer que el agente sea un buen juez sobre que acciones exploratorias son potencialmente peligrosas y cuales son acciones seguras que se pueden tomar.
- Experimentos potenciales: puede ser de ayuda tener un conjunto de entornos de juguete donde un agente incauto puede caer presa de la exploración peligrosa, pero que haya patrones de posibilidad de catástrofe suficientes para que los agentes inteligentes puedan predecirlos y evitarlos. Hasta cierto punto estas características ya existen en competiciones autónomas de helicópteros y simulaciones de rovers de Marte.

4.3. Uso malicioso de la IA

El uso malicioso de la inteligencia artificial significa un riesgo para las personas, organizaciones y estados. Puede suponer una amenaza para la seguridad digital, física y política. Existen varias propiedades relevantes en este sentido.

Los sistemas de inteligencia artificial y el conocimiento de como diseñarlos puede ser utilizado tanto constructiva como destructivamente. Los investigadores y desarrolladores no pueden elegir qué tareas producen beneficio al automatizarse y cuáles pueden ser perjudiciales, ya que en la gran mayoría de casos coexisten ambas posibilidades. Por ejemplo, un sistema que busque vulnerabilidades en el software tiene una aplicación tanto defensiva como ofensiva, y las diferencias entre las capacidades de un dron que reparta paquetes de manera automática y un dron que reparta explosivos son las mismas.

Los sistemas de inteligencia artificial suelen ser eficientes y escalables. Decimos que un sistema eficiente si una vez entrenado y desplegado puede completar cierta tarea más rápido o más barato que un humano. Escalable significa que dado un sistema que pueda completar una tarea, se pueden realizar copias del sistema que completen muchas instancias de la tarea. Un sistema de reconocimiento facial es eficiente y escalable ya que una vez desarrollado y entrenado puede

ser aplicado a diferentes cámaras por mucho menor coste que el que supondría contratar analistas humanos que hicieran el mismo trabajo.

Los sistemas de inteligencia artificial incrementan el anonimato y la distancia psicológica. Muchas tareas implican comunicación con otras personas, observar y ser observado, tomar decisiones para responder a su comportamiento o estar físicamente presente con ellos. Permitir que dichas tareas sean automatizadas puede permitir a los actores que de otra manera tendría que realizarlas conservar su anonimato y distancia psicológica de las personas a las que afectan. Una persona que utiliza un sistema de arma autónoma para cometer un asesinato en lugar de una pistola evita tener que estar presente en el lugar y tener que mirar a la víctima.

Los desarrollos de IA se prestan a una rápida difusión. Mientas que para atacantes puede ser difícil obtener o reproducir el hardware asociado a los sistemas de inteligencia artificial, es mucho más fácil obtener acceso a software y descubrimientos científicos relevantes. Muchos algoritmos de inteligencia artificial son reproducidos en cuestión de días o semanas. Además la cultura del desarrollo de la inteligencia artificial se caracteriza por un gran grado de apertura. Aunque puede ser deseable limitar la difusión de determinados desarrollos, es difícil de conseguir.

Los sistemas de inteligencia artificial actuales tienen vulnerabilidades no resueltas. Estas incluyen ataques de envenenamiento de datos (introduciendo datos de entrenamiento que causan que el sistema cometa errores), ejemplos adversos (entradas diseñadas para ser mal clasificadas por algoritmos de aprendizaje automático) y la explotación de fallos en el diseño de los objetivos de sistemas autónomos. Estas vulnerabilidades son distintas a las tradicionales del software y demuestran que los sistemas de inteligencia artificial pueden superar el rendimiento de los humanos en muchas tareas, pero también pueden fallar de maneras que un humano nunca lo haría.

5. Conclusiones

Referencias

[1] Dynamic Systems Development Method (DSDM)
http://www.students.science.uu.nl/ slegt001/me/final_5767202_Slegten.pdf

[2] Dynamic System Development Method
https://files.ifi.uzh.ch/rerg/arvo/courses/seminar_ws03/14_Voigt_DSMD_Ausarbeitung.pdf

[3] New Directions on Agile Methods:A Comparative Analysis
http://secure.com.sg/courses/ICT353/Session_Collateral/TOP_03_ART_06_ARTI-

CLE_ABRAHAMSSON_New_Directions_Agile_Methods.pdf

[4] Agile software developmentmethods.
http://www.vtt.fi/inf/pdf/publications/2002/P478.pdf

[5] Introduction to DSDM Atern
https://www.methodsandtools.com/archive/dsdmatern.php