

Medición de la habilidad cognitiva de una red neuronal artificial

¿Cómo es el desempeño de un algoritmo de inteligencia artificial en una prueba cognitiva comparado con humanos de 6-18 años?

Introducción

En esta exploración se diseñó un algoritmo de inteligencia artificial (red neuronal artificial), el cual fue entrenado con una prueba cognitiva y posteriormente probado con datos nuevos. Esta misma prueba fue aplicada a niños y adolescentes de ambos géneros. Se pretende comparar el desempeño de ambos grupos y determinar el nivel de cognición de Siegler en el que se encuentra este algoritmo.

Elegí este tema porque la inteligencia artificial es de mi interés, y es una ciencia que ha crecido mucho en los últimos años. Además, la inteligencia artificial tiene muchas aplicaciones para hoy en día y para el futuro de la humanidad, y creo que cualquier avance en esta área sería beneficioso. Pretendo analizar y dar a conocer la capacidad para resolver problemas que tienen los algoritmos de machine learning.

Antecedentes

Robert Sternberg, profesor de Psicología de IBM, define la inteligencia como “cualidad mental que consiste en las habilidades de aprender por experiencia, adaptarse a nuevas situaciones, entender y manejar conceptos abstractos, y usar el conocimiento para manipular el ambiente” (Sternberg, J. 2021).

La inteligencia es lo que nos hace únicos a algunas especies entre las otras. Los más simples de los comportamientos humanos son atribuidos a la inteligencia, mientras que el más complejo de un insecto es considerado instinto (Copeland, B. 2021). La inteligencia es una de las habilidades más importantes en nuestro desarrollo como especie y nuestra supervivencia durante el tiempo. Una de las habilidades relacionadas a esto es la cognición.

La cognición, por otro lado, es “El rango de procesos mentales relacionados a la adquisición, retenimiento, manipulación, y recuperación de la información” (Cambridge Cognition, 2015).

Los humanos tenemos alrededor de 100 billones de neuronas dedicadas para la cognición, cada una con hasta 100,000 conectores. La cognición se divide en cinco áreas principales: cognición social, función ejecutiva, memoria, atención, y velocidad psicomotor (Cambridge Cognition, 2015).

Muchos modelos de inteligencia artificial se centran principalmente en la función ejecutiva, que se refiere a la habilidad de adaptar el comportamiento, planear estratégicamente, resolver problemas, entre otras cosas.

Este estudio medirá este aspecto de la capacidad cognitiva, enfocándose en la adaptación del comportamiento, es decir, a la adaptación de la predicción de resultados basados en las etiquetas de los datos.

Según Copeland, B. (2021) La inteligencia artificial es la habilidad de una computadora de completar tareas asociadas con seres inteligentes. Involucra habilidades como la habilidad para razonar, descubrir significado, generalizar o aprender de experiencias pasadas.

Las computadoras han sido programadas para resolver problemas complejos como probar teoremas matemáticos o jugar ajedrez, muchas veces siendo incluso mejores que los humanos en áreas específicas. Por esto sus aplicaciones van desde predicción del clima, minería de datos e innovación en la industria (Borana, J. 2016), hasta diagnósticos médicos (Copeland, B. 2021).

El primer programa inteligente fue creado en 1955 por Herbert Simon y Allen Newel. Probó 38 de los 52 teoremas de *Principia Mathematica* de Russell y Whitehead (Press, G. 2016)

La inteligencia artificial se centra en cinco componentes principales de la inteligencia: aprendizaje, razonamiento, resolución de problemas, percepción, y uso del lenguaje (Copeland, B. 2016). El modelo de inteligencia artificial usado en este estudio está enfocado principalmente en el aprendizaje y el razonamiento.

Dentro de la disciplina de la inteligencia artificial está el *Machine Learning*. Es un tipo de inteligencia artificial en el que un modelo matemático aprende en base a datos.

El aprendizaje puede ser supervisado, no supervisado, semi-supervisado o por refuerzo (Editores de IBM Cloud Education, 2020). En el aprendizaje supervisado, el modelo es presentado con datos que contienen etiquetas, por ejemplo, imágenes con la etiqueta de “perro” y “gato”. En el aprendizaje no supervisado, el modelo es presentado con datos sin clasificar y los separa en grupos dependiendo de sus atributos. El aprendizaje semi-supervisado es un punto medio entre los dos anteriores en el que solo un porcentaje de los datos tienen etiquetas. Por último, en el aprendizaje por refuerzo, el modelo no tiene datos iniciales y aprende sobre la marcha según los datos que reúna en el escenario que se encuentre (Loukas, S. 2020). El modelo de este trabajo es un modelo de aprendizaje supervisado.

Los algoritmos de *Machine Learning* constan de tres partes principales: el proceso de decisión, la función de error, y el optimizador del modelo (IBM, 2020).

Dentro de la disciplina de *Machine Learning*, existen diferentes algoritmos que tienen diferentes propósitos y niveles de complejidad. El algoritmo se usará en este trabajo es una red neuronal artificial.

Una red neuronal artificial es un algoritmo complejo inspirado en el funcionamiento del cerebro humano. Consiste en miles o incluso millones de unidades de procesamiento de información (llamadas neuronas o nodos) interconectadas y organizadas en capas.

Una neurona recibe información de las neuronas de la capa anterior, la procesa y la manda a las neuronas de la siguiente capa. La información consiste generalmente de datos etiquetados a mano.

Una de las aplicaciones más populares de este algoritmo es la clasificación, pero ha sido usado con fines desde el procesamiento del lenguaje hasta reconocimiento de imágenes (Hardesty, L. 2017)

La manera en la que este algoritmo “aprende” es modificando sus pesos. Los pesos son los valores dentro del programa que representan qué tan importantes son las relaciones entre cada neurona. Éstos son modificados en cada iteración según las funciones de error y optimización.

La inteligencia y la cognición se miden de diferentes maneras.

Stenberg (2021) dice que “Una persona más inteligente representa información más claramente y opera más rápidamente sobre estas representaciones”.

Varios psicólogos han desarrollado modelos y tests para medir la inteligencia, el más popular siendo los exámenes de cociente intelectual (IQ). Un estudio realizado por Liu, Y. He, F. Zhang, H. et al. para la conferencia colectiva internacional de inteligencia artificial en 2019, midió la capacidad de distintos modelos de inteligencia artificial para resolver problemas en un test de cociente intelectual basado en 10,000 ejemplos etiquetados. El algoritmo que tuvo mejor desempeño fue una red neuronal artificial, con alrededor del 30% de respuestas correctas —más de la mitad de las de los humanos, que promediaron alrededor de 55%.

La cognición también puede ser medida con tests psicológicos. A pesar de haber varios estudios que prueban modelos de inteligencia artificial en tests de inteligencia, a día de hoy no hay estudios que midan la habilidad de un algoritmo de inteligencia artificial en una prueba cognitiva.

Tomando en cuenta que la inteligencia artificial todavía no se desarrolla hasta ser más inteligente que el adulto promedio, este estudio examinará la habilidad cognitiva de una red neuronal comparada con niños y adolescentes de ambos sexos para establecer una comparación relevante. Para esto se utilizará una modificación de la prueba cognitiva desarrollada por Robert Siegler en su estudio *Three Aspects of Cognitive Development* (Siegler, R. 1976)

Conceptos matemáticos

Esta investigación consta de dos partes principales: la predicción de outputs por medio de una red neuronal artificial que encuentra relaciones entre datos, y el

análisis y comparación de los datos de los participantes y el modelo. Para esto se utilizarán fundamentos matemáticos de aritmética, cálculo, y estadística.

Las redes neuronales aprenden por medio de la propagación hacia adelante y la propagación hacia atrás (Singh, L., 2021).

La primera, que sirve para propagar y procesar los datos en todos los nodos del modelo, utiliza funciones y la multiplicación de matrices y vectores.

Funciones de activación

Las funciones son representaciones de relaciones entre datos. Asignan cada elemento de X a uno de Y según una regla determinada. Generalmente se escriben con las letras f , g y h .

Las neuronas en una red neuronal calculan la suma de todos los datos que le llegan y sus pesos. Estos valores pueden ser cualquier valor desde $-\infty$ hasta ∞ , lo que hace imposible que las neuronas sepan si activarse o no. Las funciones de activación sirven para introducir no-linealidad en el modelo y transformar los inputs originales en unos que sirvan para determinar si la neurona es activada o no.

Las funciones que son más comúnmente usadas en las redes neuronales son ReLU y sigmoide.

La función sigmoide tiene un dominio infinito y un rango del 0 al 1, lo que la hace apta para representar probabilidades. Sin embargo, usa más recursos computacionales que la función ReLU porque es más compleja. La función sigmoide está definida por la fórmula $f(x) = \frac{1}{1+e^{-x}}$, a veces escrita como $f(x) = \frac{e^x}{e^x+1}$.

ReLU es una función de activación que simplemente regresa la parte positiva de su argumento. Utiliza muy pocos recursos computacionales y facilita el descenso

por gradiente, lo que la ha vuelto la función de activación más popular en las redes neuronales (Bharat, K., 2020). Su fórmula es $f(x) = \max(0, x)$, donde x es el input de una neurona.

Este trabajo también utiliza la función de activación “Softmax”. Se utiliza en redes neuronales de clasificación con múltiples clases, en las que la última capa tiene múltiples nodos. Cuando una red neuronal tiene múltiples nodos en su última capa y cada nodo representa una clase, es útil representar cada clase como una probabilidad.

Softmax convierte un vector de K valores reales a un vector de K valores reales entre 0 y 1. Al sumar todos estos valores siempre se obtiene 1.0

Su fórmula es

$$\sigma(\bar{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$$

Donde \bar{z} es el vector de outputs, K el número de valores en el vector, y $j = 1$ un normalizador.

Multiplicación de matrices

En las redes neuronales, los pesos se representan usando matrices. Estas son muy útiles para unir capas de neuronas y procesar sus valores.

Una matriz es un arreglo bidimensional de números. Dos matrices pueden sumarse o multiplicarse (si cumplen con los requisitos)

Se representan con letras mayúsculas como (A, B), y sus elementos con letras minúsculas y un doble subíndice (a_{11} , a_{12} , a_{13} , ...).

Dos matrices solo pueden multiplicarse si la matriz A tiene el mismo número de columnas que el número de filas de la matriz B . Si $A = [a_{ij}]$ es una matriz de dimensiones $m * n$ y $B = [b_{ij}]$ es una matriz de dimensiones $n * p$, el producto AB es una matriz de $m * p$.

El valor resultante, es decir, la matriz C , es determinado al multiplicar cada número de la fila A por cada número de la columna B .

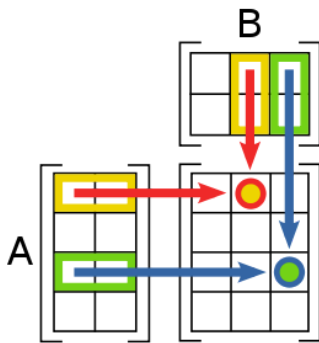


Figura 1. Diagrama que ilustra el producto de dos matrices y dando como resultado la matriz. Cortesía de Wikimedia Commons, disponible en https://commons.wikimedia.org/wiki/File:Matrix_multiplication_diagram.svg

La propagación hacia atrás es más compleja. Sirve para calcular el error de la predicción del modelo y usar ese valor para ajustar algunos de sus parámetros. Utiliza principalmente funciones de pérdida, derivadas, gradientes y algoritmos de optimización.

Funciones de pérdida y de costo

Las funciones de pérdida sirven para determinar la diferencia entre el valor predicho \hat{y} y el valor real y para cada valor. Un resultado pequeño significa que la predicción estuvo cerca del valor real.

Las funciones de costo, por su parte, son la media de los resultados de cada función de pérdida, y sirven para evaluar el rendimiento de un modelo.

Las dos funciones de pérdida más comunes en la disciplina del aprendizaje automático son error cuadrático medio y entropía cruzada. La primera es usada para problemas de regresión y la segunda para problemas de clasificación.

El error cuadrado medio mide la media de todos los errores. Su fórmula es

$$ECM = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}, \text{ donde } n \text{ es el número de pares de predicciones y valores.}$$

En el aprendizaje automático, la función de entropía cruzada es $-\log(p)$, donde p es la probabilidad predicha para la clase original. Por ejemplo, si una red neuronal clasificadora de imágenes predice que una imagen de un perro tiene una probabilidad del 52% de ser un perro, la entropía cruzada, es decir, el error, sería de $-\ln(0.52) = 0.65$.

Usar entropía cruzada en problemas de clasificación resulta en derivadas más empinadas para errores grandes, lo que le permite a la red neuronal realizar correcciones más grandes. Entre más se aleje la predicción del valor real, el valor de la pérdida sube exponencialmente

La entropía cruzada binaria es una fórmula similar que se usa para modelos que solo tienen dos clases.

Derivadas

La derivada de una función mide la sensibilidad al cambio en una función determinada, es decir, qué tanto un cambio en su argumento afecta su valor.

El cálculo de derivadas se basa en conceptos de trigonometría. Para calcular la pendiente en un punto específico de la gráfica, se unen con otro punto de la gráfica usando un triángulo. La hipotenusa de este triángulo representa la distancia entre esos puntos y, por lo tanto, una aproximación de la pendiente. Entre más infinitamente pequeños sean los triángulos, más exacto será el valor de la pendiente en ese punto.

Este proceso se puede representar mediante fórmulas matemáticas, que varían según la clase de función. Algunas de estas son:

- La derivada de un coeficiente que no va acompañado de una variable siempre es 0
- La derivada de un término ax^n con una constante, una variable, y un exponente es $n * ax^{n-1}$
- En productos de dos funciones u, v , la derivada está dada por $u * v' + v * u'$, donde u', v' son las respectivas derivadas de las funciones
- En división de dos funciones u, v , la derivada es determinada por la fórmula

$$\frac{v*u' + u*v'}{v^2}$$

Las derivadas y la diferenciación son usadas para determinar la relación entre el error y las variables de la red (pesos y bases), y usar esa información para realizar los ajustes pertinentes.

Este proceso también requiere el uso de derivadas parciales y gradientes, pero estos temas no serán cubiertos en la exploración debido a que están fuera del alcance del curso.

Algoritmos de optimización

Una vez que se conocen los errores de cada predicción y las derivadas y gradientes de cada variable, esta información es usada para optimizar el modelo. El algoritmo más común para este fin es el descenso por gradiente.

El descenso por gradiente modifica sus parámetros iterativamente para encontrar el mínimo local de una función diferenciable, es decir, se usa para encontrar los coeficientes en los que la función de error es mínima.

Los conceptos matemáticos del descenso por gradiente están fuera del alcance del curso. Sin embargo, esto no imposibilitó esta exploración porque muchas librerías de programación tienen este algoritmo implementado.

Estadística

El modelo matemático es el enfoque de esta exploración, pero también se va a usar teoría de probabilidad, media aritmética, y gráficas para analizar y comparar los datos de los participantes del estudio con el modelo.

Recolección y análisis de datos

La prueba cognitiva de Siegler fue modificada para incluir solamente 14 preguntas y se adaptó a un formato digital (fig 2). Los participantes debían predecir hacia dónde

se inclinaría una balanza según el peso que tenga en cada lado, y la distancia entre los pesos y el centro. Fue aplicada por mi a cinco grupos conformados por 10 individuos de ambos sexos.

- Grupo A: 6 a 7 años
- Grupo B: 8 a 9 años
- Grupo C: 11 a 12 años
- Grupo D: 14 a 15 años
- Grupo E: 17 a 18 años

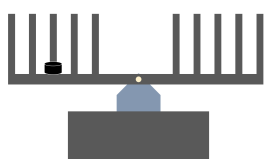
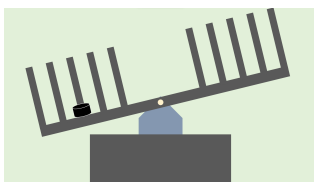
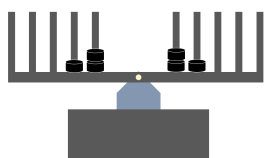
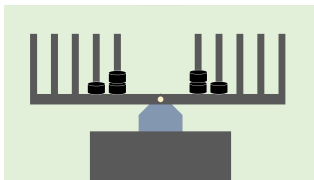
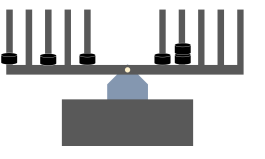
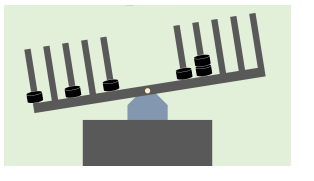
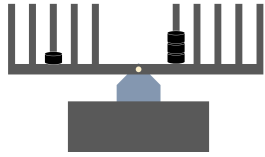
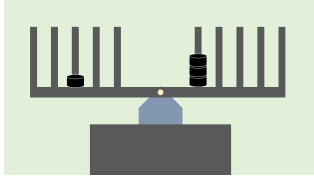
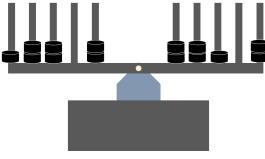
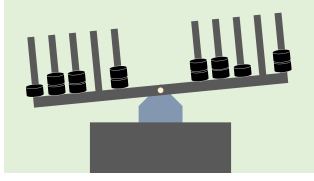
Número de pregunta	Pregunta	Respuesta
1		
5		
9		
11		
14		

Figura 2: Cinco preguntas de test con su respuesta

Después de responder cada pregunta se mostraba la respuesta correcta, como en el test original. Además, en el primer error cometido por los participantes, se les explicó a los participantes por qué su respuesta no era correcta. Esto fue una de las modificaciones y tiene el fin de incluir en los resultados la capacidad humana del aprendizaje social y hacer la comparación de inteligencias más completa.

La red neuronal fue entrenada con 626 ejemplos de balanzas con diferentes pesos en diferentes distancias y sus respectivas clases, es decir, su inclinación. Estos datos fueron recuperados de la base de datos de la Universidad de California en Irvine.

Los datos fueron revueltos aleatoriamente y luego divididos en dos grupos (entrenamiento y evaluación) con porcentajes de 70% y 30%. Los datos de entrenamiento tienen el objetivo de entrenar a la red neuronal y modificar sus parámetros, y los de evaluación tienen el objetivo de conocer la efectividad de la red neuronal en predecir las clases de nuevos datos.

Los datos fueron procesados durante 300 iteraciones. La exactitud del modelo depende de cómo fueron revueltos y divididos los datos, y alcanza valores del 93% al 99%.

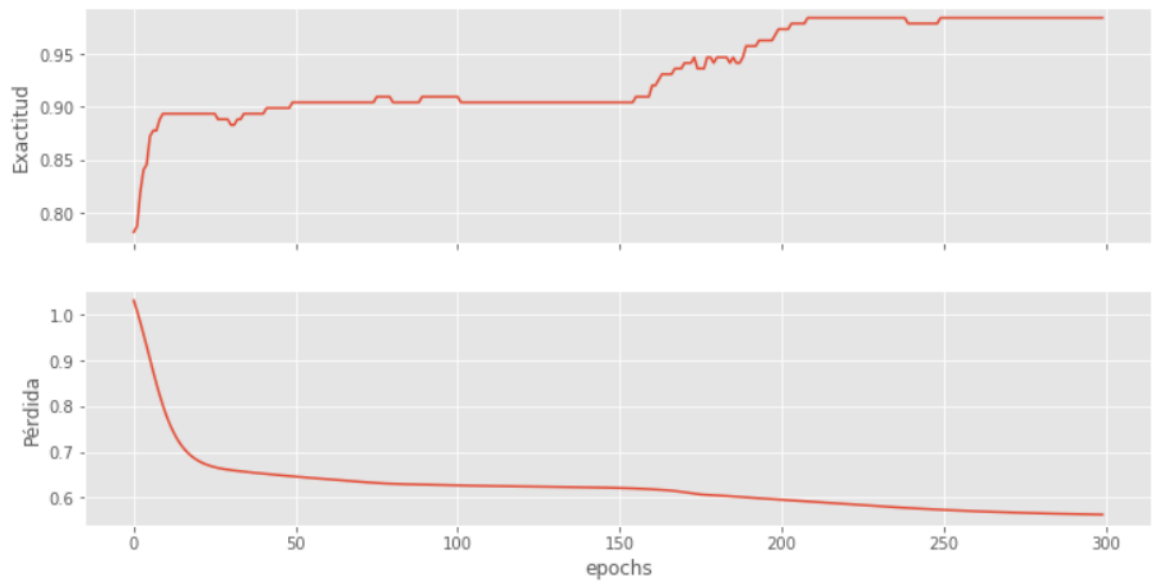


Figura 3: Gráficas que representan la exactitud y la pérdida del modelo durante 300 iteraciones

La red neuronal que se usó para la exploración consta de tres capas: la capa de input con cuatro nodos, la capa oculta con 10 nodos, y la capa de output con tres nodos.

La capa de input tiene el número de nodos de las características de los datos (Peso izquierda, distancia del peso izquierda, peso derecha, distancia del peso derecha). Por su parte, la capa de output tiene el número de nodos igual al número de clases (izquierda, centro, derecha). Los 10 nodos en la capa oculta se eligieron experimentalmente; se probaron diferentes valores y se determinó que 10 es el número óptimo de nodos para este problema.

Se utilizó como algoritmo de optimización “Estimación de Momento Adaptivo” (ADAM, por sus siglas en inglés) con una tasa de aprendizaje de 0.2. Como función de pérdida se usó “Entropía Cruzada”. Las dos primeras capas usan como función de activación “Unidad Rectificadora Lineal” (ReLU) y la última capa utiliza “Softmax”.

Por cuestiones de simplicidad, todos los cálculos están incluidos en el código anexo y no se escribieron manualmente por su cantidad, pero para demostración haré uno de los cálculos manualmente.

Cálculos

Input: [2, 0, 3, 2]

Output esperado: 2 (se inclina hacia la derecha)

Pesos de la red neuronal: [[1, 0, 2, 3, -1, 2, 2, 3, 5, 0], [2, 3, 1, 1, 0, 7, 4, 2, -2, 3]]

Output = $\text{Softmax}(\sum(\text{peso} * \text{Relu}(\text{input} * \text{peso})))$

Input 1, neurona 1, output 1 = $2 * \text{Relu}(2 * 1) = 4$

Input 1, neurona 1, output 2 = $3 * \text{Relu}(2 * 1) = 6$

Input 1, neurona 1, output 3 = $1 * \text{Relu}(2 * 1) = 2$

Softmax input 1 = $\sigma(\overline{4, 6, 2}) = \frac{e^{z_i}}{\sum_1 e^{z_j}} \approx 0.117, 0.866, 0.015$

* En las redes neuronales el softmax es calculado sólo cuando ya fueron procesados todos los inputs, pero lo calcule sobre un solo input por cuestiones de simplicidad

Output = 0.015% de probabilidad de que sea derecha (clase esperada)

Error = $-\log(0.015) \approx 1.823$

* La optimización no puede ser calculada con los valores de solamente un input

Análisis de resultados

El grupo A fue el más bajo en cuanto a respuestas correctas y el D y el E los más altos, con un promedio de 8 (57%) y 10.4 (74%) de respuestas correctas respectivamente.

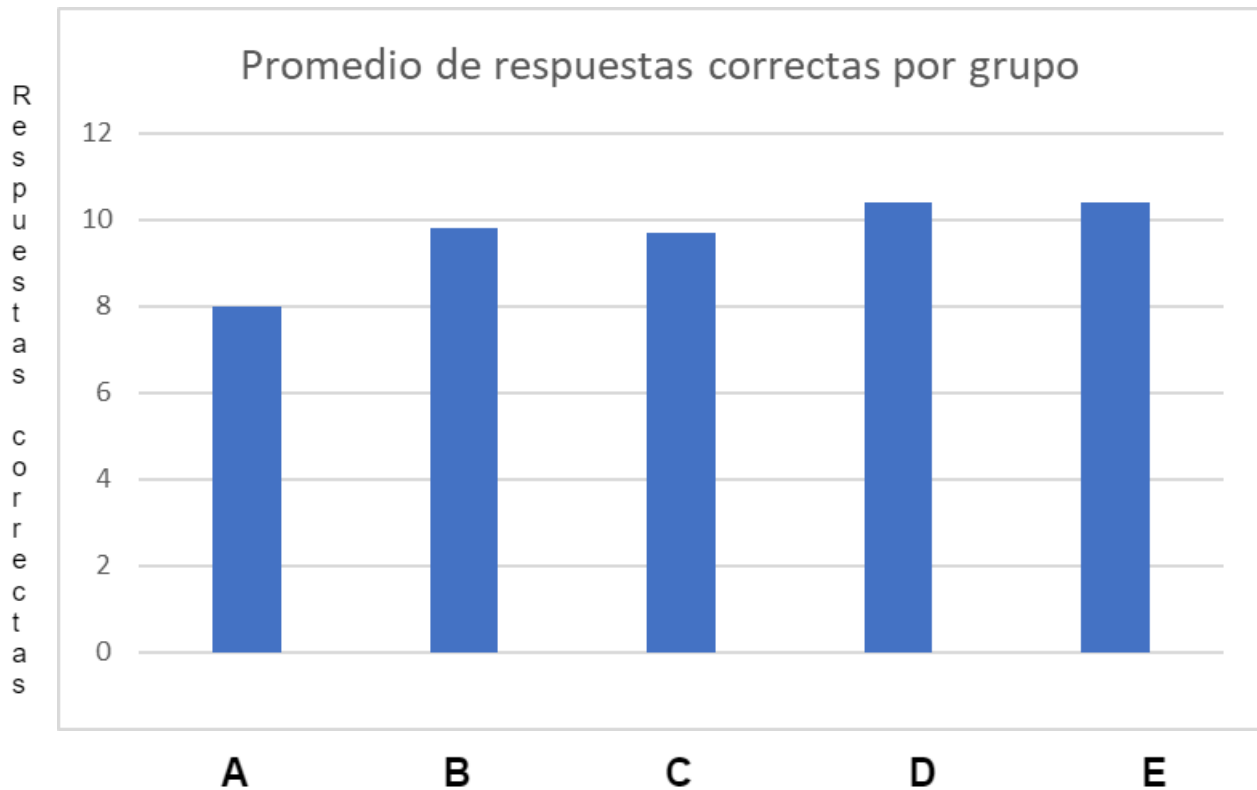


Figura 4. Promedio de respuestas correctas por grupo (A, B, C, D, E)

En promedio, los cinco grupos alcanzaron un porcentaje de respuestas correctas de 69%. Por otro lado, la red neuronal, probada en una cantidad mayor de datos, llega a alcanzar cerca del 99% de respuestas correctas.

Las preguntas de la prueba fueron diseñadas para subir gradualmente su dificultad. Cinco de las 14 preguntas tuvieron menos del 50% de aciertos, y la pregunta con menos respuestas correctas fue la 11.

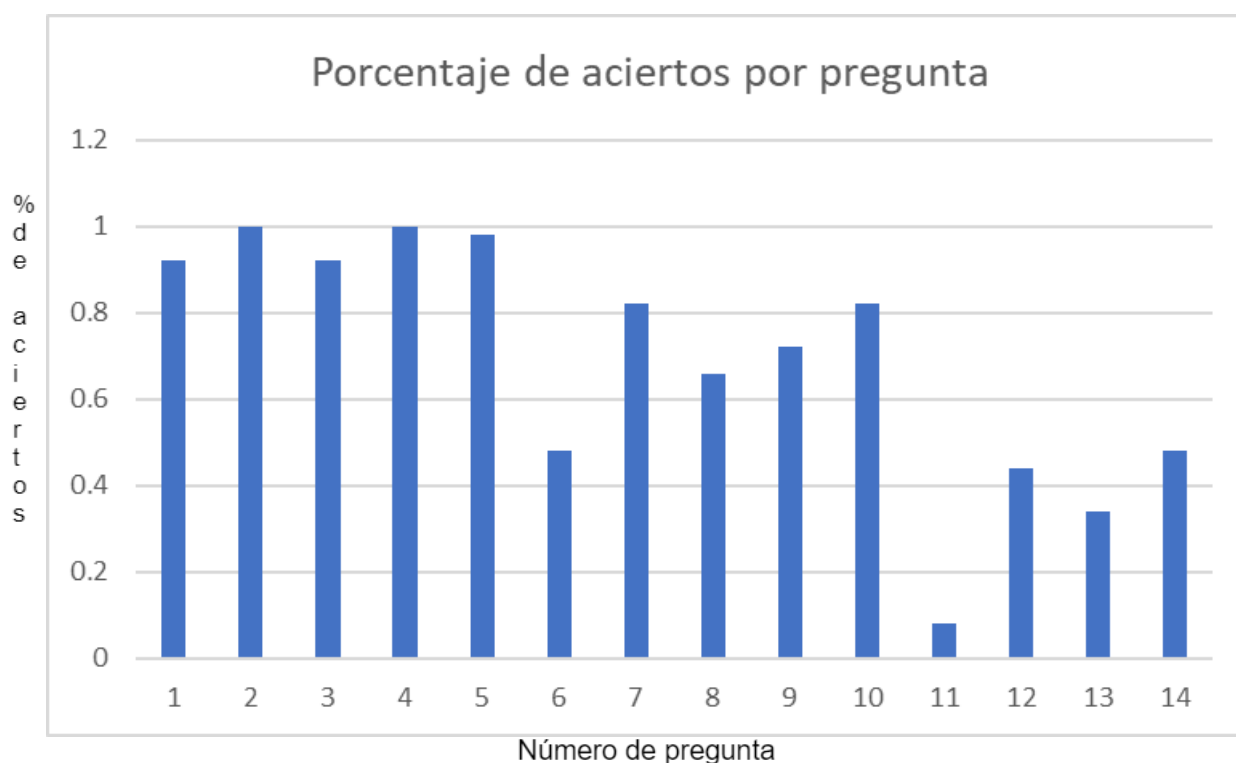


Figura 5. Porcentaje de aciertos por pregunta (promedio de los cinco grupos)

Siegler habla de etapas de desarrollo en las que los niños desarrollan reglas generales para responder estas preguntas. La primera solo considera el peso, la segunda y la tercera el peso y la distancia, y la cuarta el peso, la distancia, y el producto cruzado entre estos.

En la siguiente tabla se muestra qué regla evalúa cada una de las 14 preguntas.

Regla	Pregunta
1	1, 2, 3, 4, 5,

	10
2	6, 7, 8, 9
3	*
4	11, 12, 13, 14

*La regla 3 es ambigua. Demuestra mayor capacidad cognitiva pero no resuelve ningún problema que la regla 2 no pueda resolver

En los resultados se puede observar claramente la etapa en la que se encuentran los participantes. El integrante promedio del grupo A piensa entre el primer y el segundo nivel. Por su parte, el grupo E piensa entre el segundo y el tercer nivel, casi nunca alcanzando resultados estadísticamente significativos (mayores a la probabilidad de adivinar) en las preguntas 11-14.

El porcentaje de éxito de la red neuronal artificial la ubica en el cuarto y último nivel de cognición, sugiriendo que el modelo encontró la relación entre peso, distancia, producto cruzado, e inclinación.

Conclusión

El algoritmo de *machine learning* que se usó en el proyecto tuvo casi 25% más respuestas correctas que los adolescentes y 30% más que el participante promedio.

Se puede concluir que la red neuronal usada en el estudio es más eficiente que los niños y los adolescentes para responder esta prueba cognitiva y se encuentra en el cuatro nivel de cognición de Siegler.

El algoritmo fue capaz de encontrar relaciones entre los datos y usar estas relaciones para resolver problemas que la mayoría de los participantes humanos no pudieron.

Se cumplió el objetivo de este trabajo porque el análisis permite ver claramente la comparación entre inteligencia humana y artificial en términos de una prueba cognitiva.

Reflexión

Los resultados que obtuve en este trabajo son significativos porque se basan en las matemáticas.

En las respuestas de los humanos, use grupos de 10 personas de ambos sexos seleccionadas al azar, lo que incrementa su validez estadística. Los datos del algoritmo son determinísticos, es decir, si los mismos datos son introducidos en el mismo modelo, los resultados serán los mismos. Esto me hace creer que mis resultados son matemáticamente válidos.

En cuanto a los métodos que use, los elegí porque son métodos estandarizados en ambas áreas, recolectar datos y entrenar un modelo de machine learning. También elegí usar gráficas para representar gráficamente mis datos.

Fue un reto vincular dos áreas de la matemáticas pero creo que alcancé esto en un nivel satisfactorio, y pude usar la estadística no sólo para evaluar a los participantes humanos, también al algoritmo.

Análisis y enfoques
jpm510

Aprendí mucho acerca de realizar una investigación científica con objetividad, y pude tener experiencia aplicando temas que no se incluyen en el curso como matrices y gradientes.

Análisis y enfoques
jpm510

Anexos

Código del algoritmo, datos de entrenamiento, datos de respuesta, y prueba cognitiva disponibles en <https://github.com/MarcosSaade/CognitionModel>

Bibliografía

1. Babs, T. (2020). *The Mathematics of Neural Networks*. Medium. Recuperado enero de 2022, de <https://medium.com/coinmonks/the-mathematics-of-neural-network-60a112dd3e05>
2. Bahrath, K. (2020). *Understanding ReLU: The Most Popular Activation Function in 5 Minutes!* Medium. Recuperado enero de 2022, de <https://towardsdatascience.com/understanding-relu-the-most-popular-activation-function-in-5-minutes-459e3a2124f>
3. Billwall (chess.com). (2017). *Computers and Chess - A History*. Chess.Com. Recuperado noviembre 2021, de <https://www.chess.com/article/view/computers-and-chess---a-history>
4. Borana, J. (2016). *Applications of Artificial Intelligence & Associated Technologies*. Benemérita Universidad Autónoma de Puebla. Recuperado noviembre 2021, de https://www.cs.buap.mx/~aolvera/IA/2016_Applications%20of%20IA.pdf
5. Byju's. (2022). *Matrix Multiplication*. Recuperado enero de 2022, de <https://byjus.com/maths/matrix-multiplication/>
6. Cambridge Cognition. (2015). *What Is Cognition & Cognitive Behaviour*. Recuperado noviembre 2021, de <https://www.cambridgecognition.com/blog/entry/what-is-cognition>
7. Copeland, B. J. (2021). *Artificial Intelligence*. Encyclopedia Britannica. Recuperado noviembre 2021, de <https://www.britannica.com/technology/artificial-intelligence>

8. Dasaradh, S. K. (2021). *An Introduction To Mathematics Behind Neural Networks*. Medium. Recuperado enero de 2022, de <https://towardsdatascience.com/introduction-to-math-behind-neural-networks-e8b60dbbdeba>
9. GeeksforGeeks. (2019). *Activation Functions*. Recuperado enero de 2022, de <https://www.geeksforgeeks.org/activation-functions/>
10. Godoy, D. (2019). *Understanding binary cross-entropy / log loss: a visual explanation*. Medium. Recuperado enero de 2022, de <https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a>
11. Hardesty, L. (2017). *Explained: Neural networks*. MIT News | Massachusetts Institute of Technology. Recuperado noviembre 2021, de <https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414>
12. IBM Cloud Education. (2021). *Machine Learning*. IBM. Recuperado noviembre 2021, de <https://www.ibm.com/cloud/learn/machine-learning>
13. Liu, Y., He, F., Zhang, H., et. al. (2019). *How Well Do Machines Perform on IQ tests: a Comparison Study on a Large-Scale Dataset*. Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19), 6110–6116. <https://www.ijcai.org/proceedings/2019/0846.pdf>
14. Loukas, S. (2021). *What is Machine Learning: Supervised, Unsupervised, Semi-Supervised and Reinforcement learning methods*. Medium. Recuperado noviembre 2021, de <https://towardsdatascience.com/what-is-machine-learning-a-short-note-on-supervised-unsupervised-semi-supervised-and-aed1573ae9bb>

15. Press, G. (2016). *A Very Short History Of Artificial Intelligence (AI)*. Forbes.
Recuperado noviembre 2021, de
<https://www.forbes.com/sites/gilpress/2016/12/30/a-very-short-history-of-artificial-intelligence-ai/#:~:text=December%201955%20Herbert%20Simon%20and,Whitehead%20and%20Russell's%20Principia%20Mathematica.>
16. Probability Course. *Mean Squared Error (MSE)*. Recuperado enero de 2022, de
https://www.probabilitycourse.com/chapter9/9_1_5_mean_squared_error_MSE.php
17. Shah, S. (2021). *Cost Function | Types of Cost Function Machine Learning*. Analytics Vidhya. Recuperado enero de 2022, de
<https://www.analyticsvidhya.com/blog/2021/02/cost-function-is-no-rocket-science/>
18. Siegler, R. S. (1976). Three aspects of cognitive development. *Cognitive Psychology*, 8(4), 481–520. [https://doi.org/10.1016/0010-0285\(76\)90016-5](https://doi.org/10.1016/0010-0285(76)90016-5)
19. Singh, L. (2021). *Forward and Backward Propagation — Understanding it to master the model training process*. Medium. Recuperado enero de 2022, de
<https://medium.com/geekculture/forward-and-backward-propagation-understanding-it-to-master-the-model-training-process-3819727dc5c1>
20. StatQuest with Josh Starmer. (2021, 1 marzo). *Neural Networks Part 6: Cross Entropy* [Video]. YouTube. <https://www.youtube.com/watch?v=6ArSys5qHAU>
21. Sternberg, R. J. (1998). *Human Intelligence | Definition, Types, Test, Theories, & Facts*. Encyclopedia Britannica. Recuperado noviembre 2021, de
<https://www.britannica.com/science/human-intelligence-psychology>

22. Web Formulas. (2022). *Matrix Multiplication*. Recuperado enero de 2022, de https://www.web-formulas.com/Math_Formulas/Linear_Algebra_Matrix_Multiplication.aspx
23. Wikipedia. (2021). *Sigmoid function*. Recuperado enero de 2022, de https://en.wikipedia.org/wiki/Sigmoid_function
24. Wikipedia. (2022). *Rectifier (neural networks)*. Recuperado enero de 2022, de [https://en.wikipedia.org/wiki/Rectifier_\(neural_networks\)](https://en.wikipedia.org/wiki/Rectifier_(neural_networks))
25. Wikipedia. (2022a). *Derivative*. Recuperado enero de 2022, de <https://en.wikipedia.org/wiki/Derivative>
26. Wood, T. (2020). *Softmax Function*. DeepAI. Recuperado noviembre de 2021, de <https://deepai.org/machine-learning-glossary-and-terms/softmax-layer>