¿Es Posible Saber Deep Learning Sin Ser Ingeniero?

Fundamentos y Reflexiones sobre Modelos Generativos

Ana Guerrero Tamayo
Fernando José Sadio-Ramos

Universidad de Deusto
Institulo Politécnico de Coimbra

Ficha Técnica

ISBN: 978-989-8486-52-3
Título: ¿Es Posible Saber Deep Learning Sin Ser Ingeniero? Fundamentos y Reflexiones sobre Modelos Generativos
Autores: Ana Guerrero Tamayo; Fernando José Sadio-Ramos
Edición: Fernando Ramos (Editor)® (Coimbra)
Layout y composición:: Ana Guerrero Tamayo
CDU: 001; 004.8; 17; 342.7; 37

Apoyo

O Instituto de Estudos Filosóficos é financiado por fundos nacionais através da FCT – Fundação para a Ciência e a Tecnologia, I.P., no âmbito do projeto UID/FIL/00010/2020

Dedicado a todos los que nos ayudan a ser cada día un poquito mejores...

Agradecimientos

A la persona que está leyendo ahora mismo estas líneas: muchísimas gracias por elegirnos. Gracias por darnos la oportunidad. Y esperamos que usted encuentre por aquí un poquito de lo que busca. Y si no encuentra lo que busca, que al menos encuentre lo que necesite.

Queremos agradecer su colaboración y apoyo a la Profesora Doctora María Angustias Ortíz Molina. Gracias por escucharnos, por todas tus opiniones, por todas esas palabras amables. Gracias por todo el trabajo que sigues haciendo. Gracias por tu trayectoria, por tu conocimiento y por tu generosidad divulgándolo. Gracias porque con tu trabajo has conseguido mejorar este caos de mundo.

Queremos agradecer asimismo todo su apoyo a los Professores Doutores João Paulo Martins Gouveia y Teresa Cristina Melo Fragoso. Gracias a vosotros hoy hemos escrito este libro. ¡Qué grandísima suerte haberos conocido!

Gracias a Izaskun Kintana Arkoitza por llenar de música la escritura de este libro. Y gracias por demostrar lo pequeño que es a veces el infinito...

Gracias al Profesor Doctor Héctor Archilla Segade y a Eunate Izaola Ibáñez por ser unos grandes amigos de los que quedan pocos.

Gracias a nuestra-tímida-amiga-que-no-quiere-figurar B. Gracias por ser esa amiga que todos deberíamos tener al lado. Gracias por ser esa persona que, cuando todos se van, ella viene a quedarse.

Gracias a I. y a M. por sacar las mejores fotos de patos. Y por habernos llenado de cosas bonitas el corazón.

Gracias a toda esa familia y amigos que siempre nos levantan cuando nos caemos y nos protegen cuando lo necesitamos.

Gracias a nuestros angelitos Mario, María, Nereia, Talia, Amélie, Izatz, Umi, Yoru, Sora, Noa, Runa, Serkan, Eda, Juno. Gracias por aterrizar en nuestras vidas para enseñarnos lo que es el amor verdadero...

Mario, Runa, Serkan, Izatz, Umi, qué guapos salís en la foto...

Prefacio

Qué libro más extraño...

Nos hemos juntado una ingeniera electrónica y un filósofo y hemos escrito sobre Computación, sobre Psicología, sobre Sociología y sobre Derecho. ¡Toma ya!

Cualquiera podría pensar que somos unos sinvergüenzas que hemos escrito un libro para hablar de cosas de las que no tenemos ni idea.

Puede incluso que sea cierto (lo de sinvergüenzas, lo otro no).

Somos una ingeniera electrónica y un filósofo. Somos diferentes, muuuy diferentes. Pero coincidimos en lo importante:

1. Tenemos sentido común. No sabemos si tenemos mucho o poco. De momento, con tenerlo es suficiente.

2. Hemos leído muchísimo y de todo.

3. La vida nos ha obligado en repetidas ocasiones a aprender rápido y bien de ámbitos totalmente desconocidos para nosotros.

4. Nos divierte mucho observar y pensar.

5. Lo políticamente correcto nos aburre una barbaridad.

Así que nuestras diferencias en todo lo no importante nos han permitido complementarnos para tener una visión holística de este asunto de los modelos generativos.

Y ofrecer nuestra visión, que de eso se trata.

No esperen encontrar en este libro:

1. Un manual de referencia. Nosotros no somos expertos de nada, no tenemos tanto ego. Como dijo Oscar Wilde, no somos tan jóvenes como para saberlo todo. Simplemente queremos que cualquier persona que lea este libro se maneje suficientemente bien en cuanto a trabajar con modelos Machine Learning y Deep Learning.

2. Textos políticamente correctos. Al pan, pan y al vino, vino. Es muy desagradable leer textos sobre ética, moral, problemas sociales, etc. que pasan por encima de todo, que aportan generalidades, que no ponen los problemas tal y como son encima de la mesa. Que no pisan charcos. Que son políticamente correctos. Nosotros somos lo suficientemente torpes para pisar los charcos que haga falta y llamar a las cosas por su nombre. Y procuramos hacerlo con responsabilidad y con algo de valentía.

Si en algún momento echan de menos que pisemos con más fuerza un charco, pedimos disculpas y alegamos que tampoco nos apetece meternos en un lío.

No somos políticos pero tampoco somos revolucionarios.

Con este libro sólo queremos explicar todo esto del Machine Learning y el Deep Learning para que cualquier persona lega en computación o en ingeniería pueda manejarse un poquito en el ámbito. Quitar un poco ese halo de magia y misterio sobre estos modelos. Querido lector, si este libro se queda corto, nos hará muy felices porque habremos sido capaces de explicar algunas bases y fundamentos de forma correcta.

Y, lo más importante, queremos compartir nuestros pensamientos por si pueden hacer pensar al que nos lea, queremos invitar a hacer autocrítica, a aplicar el sentido común, ya no en el uso o el diseño de estos modelos, sino en nuestra vida. En toda ella.

Si hemos conseguido esto segundo, ahora sí que somos felices.

Personalmente, nos hemos divertido mucho pensando y escribiendo este libro. Hemos aprendido mucho de nosotros mismos.

Y ojalá podamos aportar algo para que usted, querido lector, aprenda también.

Índice General

Lista de Figuras

Abreviaturas

AUC . . . . . Area Under the Curve.

CAE . . . . . Convolutional Autoencoder.

CEO . . . . . Chief Executive Officer.

CNN . . . . . Convolutional Neural Network.

ETS . . . . . Enfermedad de Transmisión Sexual.

FPR . . . . . False Postive Rate.

GAN . . . . . Generative Adversial Network.

GPT . . . . . Generative Pre-trained Transformer.

GPU . . . . . Graphics Processing Unit.

Grad-CAM . . Gradient-weighted Class Activation Mapping.

GRU . . . . . Gated Recurrent Unit.

IA . . . . . Inteligencia Artificial.

LER . . . . . Lesiones por Esfuerzo Repetitivo.

LLM . . . . . Large Language Models.

LSTM . . . . . Long Short-Term Memory.

NLP . . . . . Natural Language Processing.

RAE . . . . . Real Academia Española.

RAE . . . . . Recurrent Autoencoder.

RAM . . . . . Random Access Memory.

ReLU . . . . . Rectified Linear Unit.

RNN . . . . . Recurrent Neural Network.

ROC . . . . . Receiver Operating Characteristic.

SGD . . . . . Stocastic Gradient Descent.

tanh . . . . . Tangente Hiperbólica.

TPR . . . . . True Positive Rate.

VAE . . . . . Variational Autoencoder.

Si me engañas una vez, tuya es la culpa; si me engañas dos, es mía

— Anaxágoras, *en el siglo V a.C. afirmó que la Luna no emite luz propia, sino que refleja la luz del Sol.*

1 Conceptos Básicos en Machine Learning

Índice

1.1. ¿Qué es Inteligencia Artificial Realmente?

Si ustedes investigan un poco en bibliografía especializada, verán que existen muchas y variopintas definiciones de Inteligencia Artificial (IA), ya desde que surge esta denominación aproximadamente en 1950.

Esta falta de concreción para definir este concepto nos hace pensar que tal vez sea porque el concepto es incorrecto... O simplemente no existe.

No dudamos que hay una componente muy fuerte de marketing en esa denominación de Inteligencia Artificial. ¿Cómo es posible? ¿'Inteligencia Artificial' es sobre todo marketing? ¿Por qué? Muy sencillo, porque NO ES INTELIGENCIA.

Esta denominación incorrecta basada en elemento de marketing es similar al concepto informático de 'tiempo real'. Control en tiempo real, visualización en tiempo real, programación en tiempo real... Otra incorrección. Si preguntamos a

un matemático o a un físico, seguramente acotará este concepto mucho más allá que lo que hoy se denomina 'tiempo real' en ingeniería.

Cualquier procesado o transmisión de datos implica necesariamente un retardo. Si nos atenemos a una definición literal del concepto 'tiempo real', que implica un retardo = 0 prácticamente, nos daremos cuenta de que la mayor parte de los procesos denominados así, realmente no lo son.

Más allá, podemos asumir como tiempo real todo aquello cuyo retardo sea ínfimo, prácticamente tendente a 0 a efectos prácticos. Sin embargo, probablemente encontraremos como argumento de venta o elección 'tiempo real' cuando realmente el procesado implica varios segundos. Marketing.

Desde el punto de vista de la ingeniería, la denominación correcta es Machine Learning. Es decir, que las máquinas aprendan. Esto sí.

Una observación: que las máquinas tengan capacidad de aprender, no quiere decir que sean inteligentes.

Estas tecnologías son algoritmos. La definición de algoritmo según la RAE es: *'1. m. Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema. 2. m. Método y notación en las distintas formas del cálculo.'*

No es nada más, no hay más misterio. Es un conjunto de operaciones matemáticas. Esa niebla mística que rodea estas magias de la IA está desapareciendo.

Indudablemente, el desarrollo de estas herramientas ha supuesto un cambio de paradigma, que se resume muy bien en la Figura 1.1:

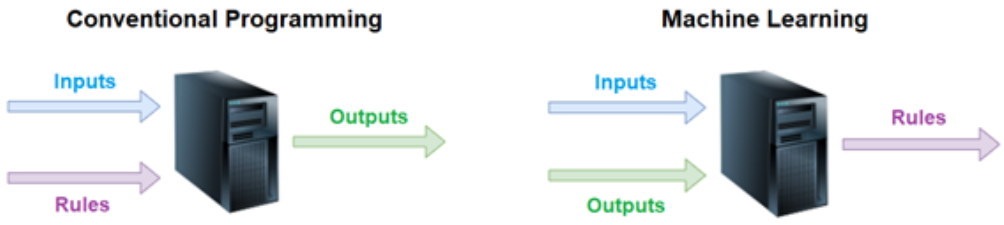


Figura 1.1: Programación Tradicional vs. Machine Learning

En la programación tradicional, tenemos unos datos de entrada, programamos un código donde decimos qué operaciones hay que hacer sobre esos datos de entrada y, con todo ello, obtenemos unos datos de salida.

El desarrollo de algoritmos Machine Learning supone un cambio sustancial en esta dinámica. Tenemos unos datos de entrada y unos datos de salida. Los algoritmos Machine Learning determinan las normas que rigen la relación entre esos datos de entrada y de salida.

Un ejemplo: un algoritmo Machine Learning que distingue perros de gatos en imágenes. Ese algoritmo 'ha visto' miles, millones de imágenes de perros y de gatos, ha detectado los patrones que establecen claramente que esa imagen es un perro o que esa imagen es un gato. Y ante una imagen nueva, dirá si es un perro o un gato. Es decir, un clasificador de perros y gatos.

Por eso decimos que las máquinas aprenden. Y esto ha supuesto un avance fundamental en el ámbito de la computación.

¡Y todo gracias a las matemáticas!

Los algoritmos Machine Learning utilizan diversas ramas de matemática aplicada de una forma compleja, sobre todo en el diseño. Tenemos álgebra (cálculo matricial), tenemos cálculo, tenemos mucha estadística y probabilidad...

El complejo diseño de estas arquitecturas es simplemente admirable. Y es aún mas admirable porque, desde la complejidad en el diseño, han conseguido que su aplicación sea sencilla. Aplicar estos diseños es fácil.

Sobre todo si lo comparamos con la complejidad de pensarlos. Eso sí que es difícil.

Y aún puede ser más difícil. Dentro de los algoritmos Machine Learning, hay una rama especialmente evolucionada denominada Deep Learning.

Siguen siendo algoritmos. Con la particularidad de que DICEN IMITAR el funcionamiento de las neuronas humanas. Volvemos a pensar inevitablemente en el marketing detrás de esta denominación.

¿Sabemos realmente cómo funcionan las neuronas, ya no humanas, sino animales? La respuesta es NO. Hemos sido capaces de intuir ciertos mecanismos de funcionamiento neuronal, pero aún desconocemos cómo funcionan las neuronas. A todos los niveles.

Por lo tanto, utilizando una lógica cartesiana muy básica, debo pensar que eso que dicen que los algoritmos Deep Learning imitan el funcionamiento de las neuronas humanas es muy inexacto.

Los algoritmos Deep Learning son arquitecturas más o menos complejas cuya unidad básica es eso que denominan 'neurona artificial'. Entre comillas.

En esta época, marzo de 2024, oímos hablar de Inteligencia Artificial Generativa constantemente. Tampoco es Inteligencia por todo lo expuesto. El término correcto es 'modelos generativos'.

Detrás de estos algoritmos, una vez más, simplemente hay algoritmos. Algoritmos muy elaborados, es cierto. Algoritmos con unas arquitecturas muy creativas, como es el caso de las redes GAN y de los mecanismos de atención en las técnicas de procesado de lenguaje natural. Aquí la inteligencia sigue siendo humana y la poseen los diseñadores. Y mucha.

Ya hemos desmitificado un poquito el 'Universo IA'. Ya hemos olfateado el marketing en la construcción del discurso en torno a estos algoritmos.

Y ahora que hemos encuadrado un poquito de qué estamos hablando realmente, vamos a entrar en harina en los siguientes capítulos.

Evidentemente, utilizaremos el concepto 'Inteligencia Artificial' lo mínimo indispensable. Tal vez el texto pierda encanto pero sin duda será más fidedigno.

1.2. ¿Qué Necesitamos para Aplicar Algoritmos Machine Learning?

Estos algoritmos se alimentan de datos para extraer esas normas, esos patrones, que rigen los datos. Por lo tanto, lo primero que necesitamos es:

1. DATOS.

Fundamental. Estos algoritmos son potentísimos identificando patrones, especialmente los algoritmos Deep Learning. Pero necesitan miles de datos. Y si son millones de datos, mejor.

Esos datos deben reunir una serie de características que aseguren que ese algoritmo funcione lo mejor posible. Reiteramos: hemos dicho exactamente LO MEJOR POSIBLE.

Con esos datos tenemos que entrenar al algoritmo. Es decir, el algoritmo, que en este momento es una página en blanco, tiene que aprender de esos datos (de ahí el nombre de Machine Learning, Deep Learning...).

Y tenemos que programar este aprendizaje. Nosotros le indicamos al algoritmo cómo tiene que aprender con esos datos de entrada y de salida para detectar correctamente las normas que les rigen.

Estas indicaciones de cómo tienen que aprender se traducen en el ajuste de una serie de hiperparámetros, algún factor adicional, etc. Y es fundamental verificar bien lo obtenido tras la finalización de ese entrenamiento.

Por tanto, lo segundo que necesitamos para aplicar estos algoritmos es:

2. UN BUEN ENTRENAMIENTO CON ESOS DATOS.

Es sorprendente que, a pesar de la base matemática detrás de estos algoritmos, la forma de verificar que el entrenamiento es correcto es puramente empírica. Ensayo-error.

Es sorprendente pero es así. Se prueban diferentes configuraciones de hiperparámetros y se selecciona aquella que cumple mejor los estándares de lo que se considera un entrenamiento correcto.

Y, por supuesto, para manejar todos estos datos, el algoritmo, etc. necesitamos:

3. UN BUEN ORDENADOR.

Especialmente las arquitecturas Deep Learning requieren mucha computación. Como además necesitan manejar miles, millones de datos, más computación aún. Y como tendremos que realizar múltiples entrenamientos para asegurar una buena elección de hiperparámetros, más computación aún.

Todo esto difícilmente puede correr en un ordenador de uso doméstico. Necesitaremos una buena GPU (Graphics Processing Unit, esto es, Unidad de Procesamiento Gráfico) si trabajamos con imágenes, mucha memoria RAM, mucho disco duro y un procesador muy potente (o varios).

Escucha, serás sabio. El comienzo de la sabiduría es el silencio

— Pitágoras, *descubrió que la longitud de las cuerdas de un instrumento musical influía en el tono que producían cuando se las tocaba. Este descubrimiento sentó las bases para la relación entre las matemáticas y la música.*

2 Algunos Conceptos Técnicos

Índice

2.1. Nota de los Autores Relativa al Uso de Anglicismos

Rogamos disculpen el uso de anglicismos en este capítulo. Tras un intenso análisis sobre la estrategia a seguir, los autores optamos por expresar los conceptos computacionales con anglicismos. Son los términos ampliamente utilizados en la mayor parte de libros de esta temática, son los términos que utilizan los

profesionales en el área. Utilizar los términos castellanizados, desgraciadamente, será probablemente una traba en la comunicación.

Así que hemos optado por incumplir el uso correcto de anglicismos en pos de que el lector se familiarice con los términos realmente utilizados en ingeniería y computación.

2.2. Tipos de Algoritmo, a Grandes Rasgos

Hay 3 grandes tipos de algoritmos en Machine Learning:

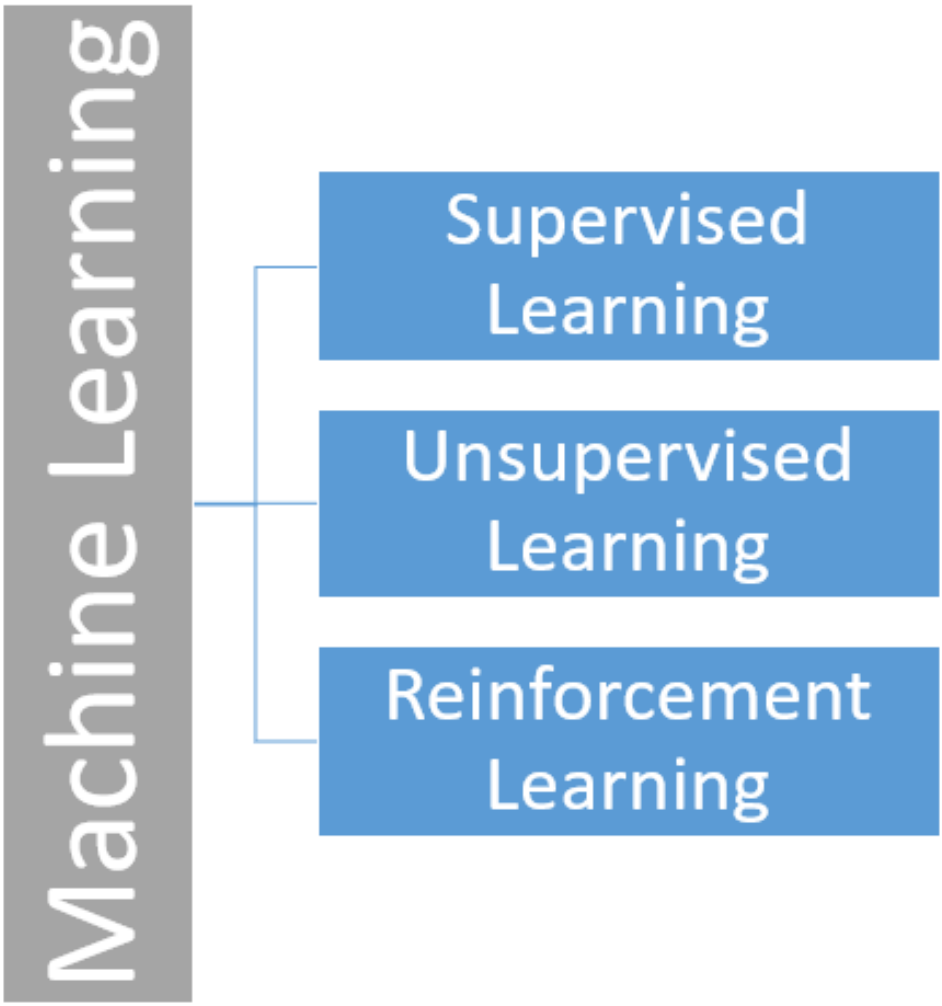


Figura 2.1: Tipos de Algoritmo

En Supervised Learning (aprendizaje supervisado), los algoritmos trabajan con

datos 'etiquetados' (labeled data).

Estos algoritmos aprenden basándose en un sistema de etiquetas asociadas a unos datos. En base a esas etiquetas aprenden y pueden posteriormente a ese aprendizaje tomar decisiones o hacer predicciones.

Muchos de estos algoritmos están dedicados a la clasificación. Se utilizan para asignar una clase o categoría a nuevos datos basándose en patrones aprendidos a partir de datos previamente etiquetados.

Un ejemplo es un detector de spam, que etiqueta un e-mail como spam o no dependiendo de los patrones que ha aprendido del histórico de correos previamente etiquetados como 'spam' o 'no spam' (remitente, relación texto/imágenes, palabras clave en el asunto, etc.).

Otro importante grupo de algoritmos se dedica a la regresión. Su función es modelar la relación entre una variable dependiente (también llamada variable de respuesta) y una o más variables independientes (también llamadas predictores o variables explicativas). El objetivo principal de la regresión es predecir o estimar el valor de la variable dependiente en función de los valores de las variables independientes.

En Unsupervised Learning (aprendizaje no supervisado), no disponemos de datos etiquetados para el entrenamiento. Por ello, tienen un carácter exploratorio.

Estos algoritmos no cuentan con un conocimiento basado en etiquetas (labels), porque esos datos no están etiquetados.

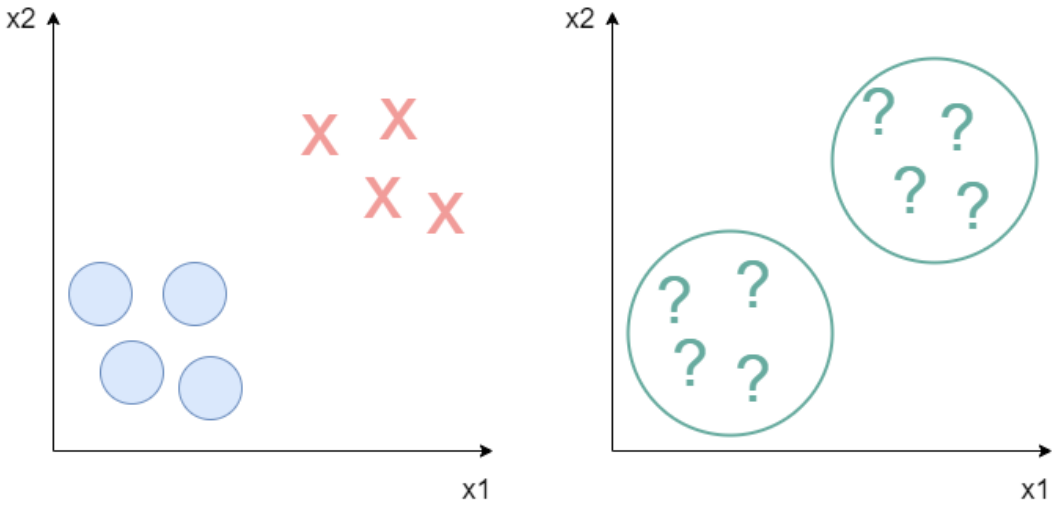
Se enfrentan al caos de datos con el objetivo de encontrar patrones que permitan organizarlos de alguna manera. Por ejemplo, en el campo del marketing se utilizan para extraer patrones de datos masivos provenientes de las redes sociales y crear campañas de publicidad altamente segmentadas.

En Unsupervised Learning, destacan los algoritmos de clustering (agrupamiento). Estos algoritmos clasifican en grupos los datos. Por ejemplo, las segmentaciones de clientes según qué hayan comprado.

Otro grupo de algoritmos dentro de Unsupervised Learning está orientados a la detección de asociaciones. Descubren reglas dentro del conjunto de datos. Por

ejemplo, aquellos clientes que compran un coche también contratan un seguro, por lo que el algoritmo detecta esta regla.

Esta figura puede aclarar la diferencia entre Supervised Learning (con labels) y Unsupervised Learning (sin labels):



(a) Supervised Learning **(b)** Unsupervised Learning

Figura 2.2: Diferencia entre Supervised Learning y Unsupervised Learning

En Supervised Learning tenemos dos grupos con sus respectivas etiquetas. Por ejemplo, un grupo de perros y otro de gatos. El algoritmo detecta ambos grupos y 'sabe' que un grupo corresponde a perros y el otro a gatos.

En Unsupervised Learning, el algoritmo sólo detecta dos grupos. No sabe si son perros o gatos porque no hay etiquetas. Sólo reconoce que son dos grupos diferentes, pero 'no sabe poner nombre' a esos grupos.

El tercer tipo, el tercer grupo de algoritmos Machine Learning es el denominado Reinforcement Learning (aprendizaje por refuerzo). Su objetivo es que un algoritmo aprenda a partir de la propia experiencia. Esto es, que sea capaz de tomar la mejor decisión ante diferentes situaciones de acuerdo a un proceso de prueba y error en el que se recompensan las decisiones correctas.

Se puede utilizar Reinforcement Learning en las siguientes situaciones:

- Se conoce un modelo del entorno, pero no se dispone de una solución analítica.

- Sólo se dispone de un modelo de simulación del entorno (objeto de la optimización basada en la simulación).

- La única forma de recopilar información sobre el entorno es interactuar con él.

2.3. Deep Learning. Principio de Diseño

Dentro del compendio de algoritmos Machine Learning destaca un subconjunto: Deep Learning.

Este subconjunto de algoritmos dentro de Machine Learning se caracteriza porque se inspira en el funcionamiento del cerebro humano. Esta inspiración, intento de imitación, le permite detectar patrones complejos en los datos de una manera muy potente.

Dado que estos algoritmos intentan imitar el funcionamiento del cerebro humano, su arquitectura se basa en redes neuronales.

Una red neuronal es un modelo de computación cuya estructura de capas se asemeja a la estructura interconectada de las neuronas en el cerebro, con capas de nodos conectados. Una red neuronal puede aprender de los datos, de manera que se puede entrenar para que reconozca patrones, clasifique datos y pronostique eventos futuros.

Las redes neuronales descomponen las entradas en capas de abstracción. Su comportamiento está definido por la forma en que se conectan sus elementos individuales, así como por la importancia (o ponderación) de dichas conexiones. Estas ponderaciones se ajustan automáticamente durante el entrenamiento de acuerdo con una regla de aprendizaje especificada hasta que la red neuronal lleva a cabo la tarea deseada correctamente.

Estas redes neuronales, formadas por capas de nodos interconectados, tienen este aspecto, a grandes rasgos:

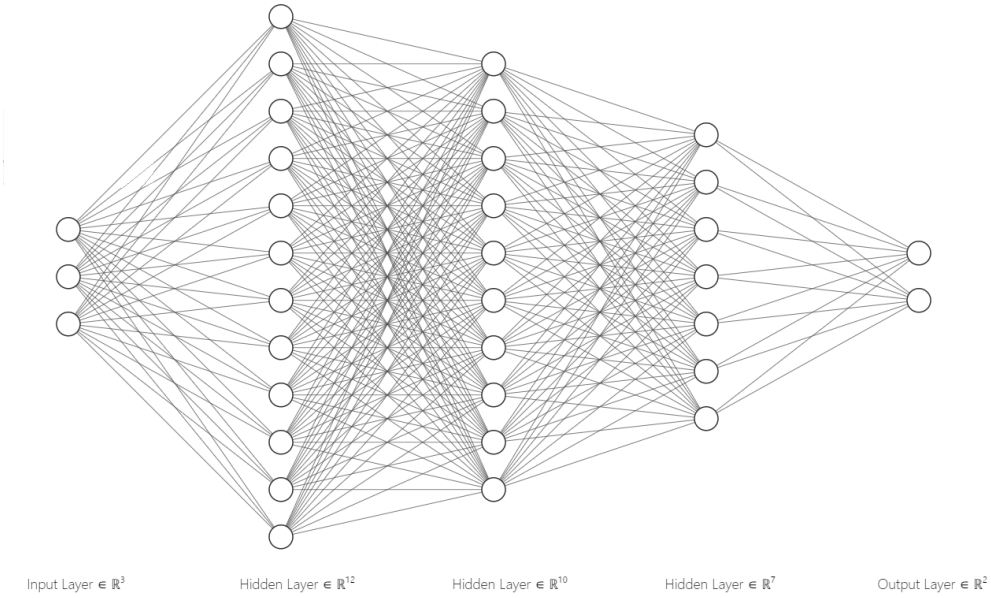


Figura 2.3: Esquema de Concepto de Red Neuronal

Cada uno de los nodos (los círculos de la red anterior) se denomina 'neurona artificial'. No nos gusta este nombre porque vuelve a ser inexacto (no sabemos cómo funciona una neurona realmente) pero no hay alternativa. Este es el nombre oficial.

La neurona artificial es la unidad de procesamiento básica de las redes neuronales artificiales. Cada neurona tiene diferentes pesos (W) y un parámetro b (bias) que es igual entre las neuronas de una misma capa.

La siguiente figura es una representación esquemática del funcionamiento de una neurona artificial, de una forma básica.

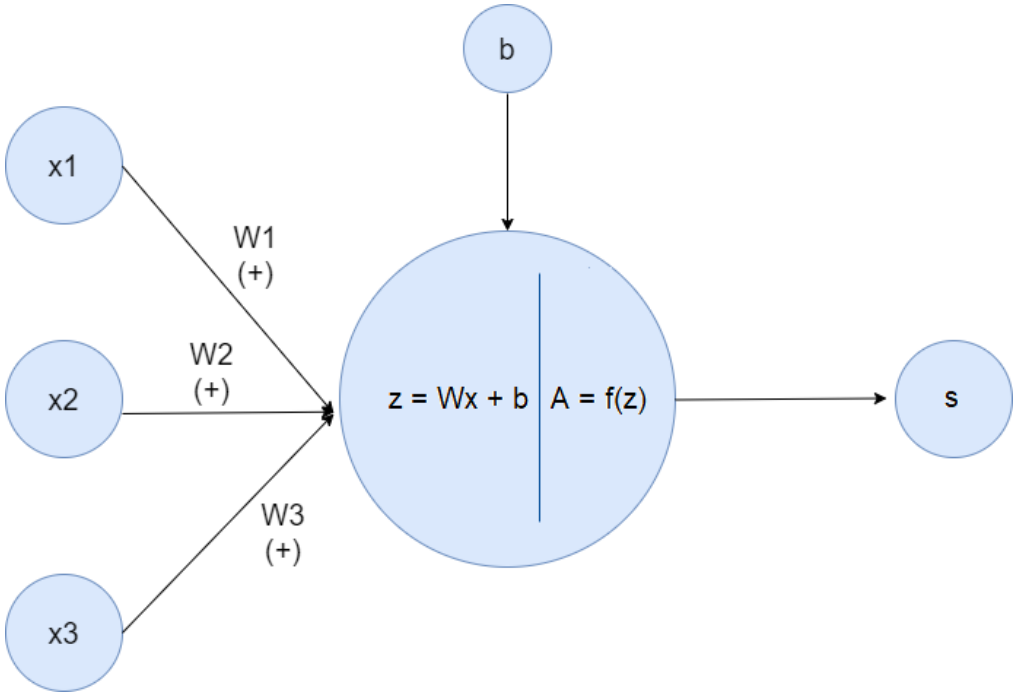


Figura 2.4: Neurona Artificial

Donde:

- A es la salida de la neurona. Se genera, por decir así, en dos etapas. En la primera se realiza una operación lineal y en la segunda se aplica la función de activación.

- W es un vector de pesos que representa la importancia relativa de cada una de las entradas, esto es [W1, W2, W3,...].

- x es el vector de entradas, esto es, [x1, x2, x3,...]

- b es el sesgo o término de sesgo (bias).Es una constante que permite mover la función de activación.

La neurona artificial realiza una operación lineal seguida de una función de activación no lineal. La operación lineal es simplemente el producto punto entre el vector de entradas y el vector de pesos, más el sesgo:

$$z = Wx + b \tag{2.1}$$

Donde:

- z es el resultado de la operación lineal. Luego, este resultado se pasa a través de una función de activación para producir la salida de la neurona:

$$A = f(z) \tag{2.2}$$

Donde A es la salida de la neurona.

La función de activación introduce no linealidades en la red neuronal, permitiendo que las redes neuronales aprendan relaciones y patrones complejos en los datos. Ejemplos comunes de funciones de activación incluyen la función sigmoide, la función de activación ReLU (Rectified Linear Unit), entre otras.

En la práctica, al diseñar la red neuronal, el diseñador simplemente elige la función de activación que más le conviene para el ámbito de aplicación del modelo diseñado.

En resumen, una neurona artificial combina las entradas ponderadas con los pesos y el sesgo, y luego aplica una función de activación para producir una salida. Este proceso se repite para todas las neuronas en una red neuronal, donde las salidas de una capa de neuronas se convierten en las entradas de la siguiente capa.

2.4. Ciclo de Entrenamiento

Para facilitar la comprensión de cómo funciona el entrenamiento de uno de estos modelos, vamos a observar este caso, un algoritmo de clasificación (Supervised Learning).

El siguiente esquema ilustra de manera gráfica el proceso de entrenamiento de un modelo:

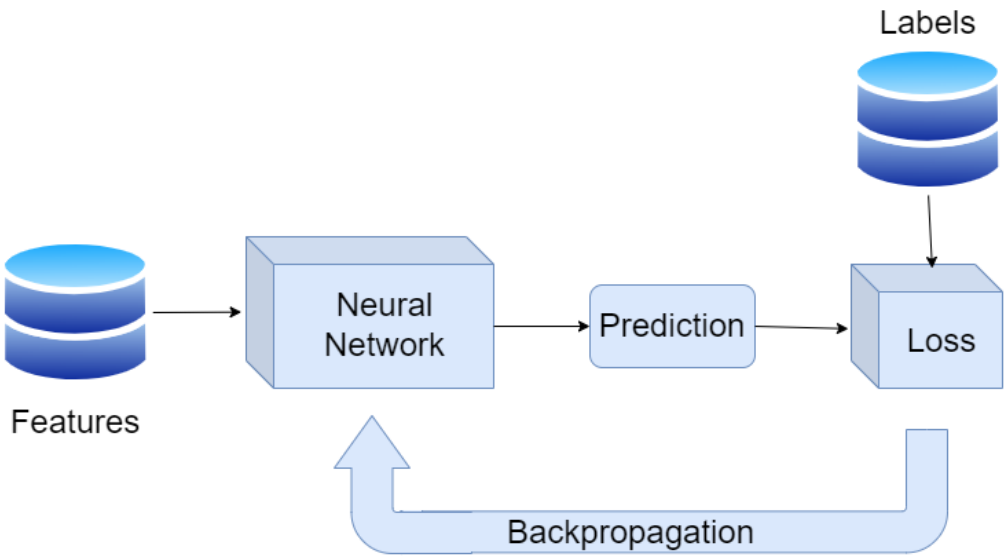


Figura 2.5: Proceso de Entrenamiento

Antes de explicar los pasos de un entrenamiento, debemos hacer una distinción entre parámetros e hiperparámetros. Son dos conceptos diferentes.

Los parámetros del modelo son las variables INTERNAS que el algoritmo de aprendizaje automático ajusta automáticamente durante el proceso de entrenamiento para minimizar la función de pérdida y hacer que el modelo se ajuste mejor a los datos de entrenamiento. Estos parámetros representan las relaciones y patrones que el modelo aprende de los datos.

Algunos ejemplos de parámetros:

- Pesos (weights): Representan la fuerza de la conexión entre neuronas en diferentes capas de la red. Cada conexión entre dos neuronas tiene asociado un peso que indica la importancia de la señal de salida de una neurona para la neurona de entrada siguiente. Es el vector W explicado en la sección anterior.

- Sesgos (biases): Son términos adicionales añadidos a las entradas de cada neurona (a menudo como una entrada adicional con un valor constante) que permite que la red aprenda funciones no lineales y desplazamientos. La b explicada en la sección anterior.

En cambio, los hiperparámetros son parámetros EXTERNOS al modelo de

aprendizaje automático que controlan el proceso de entrenamiento y la configuración del modelo.

A diferencia de los parámetros del modelo, cuyo valor se establece directamente de los datos durante el entrenamiento de forma automática, los hiperparámetros los establecemos nosotros antes de iniciar el proceso de entrenamiento. Los hiperparámetros afectarán a los valores que vayan tomando los parámetros del modelo.

Por lo tanto, actúan como configuraciones que guían el comportamiento y la complejidad del modelo durante el proceso de optimización.

Veamos los pasos básicos de este proceso de entrenamiento:

1. Preparación de los datos. Se recopilan y procesan los datos relevantes para el problema de clasificación. Cada instancia de datos suele estar representada por un conjunto de características (atributos) y su respectiva etiqueta de clase (label).

2. División de los datos. El conjunto de datos se divide en conjuntos de entrenamiento (Training Set) y validación (Validation Set) para evaluar el rendimiento del modelo. El conjunto de validación se utiliza para ajustar los hiperparámetros del modelo.

3. Definición del modelo. Se construye un modelo, su arquitectura. Por ejemplo, en el caso de una red neuronal, la construimos definiendo su número de capas, las neuronas de cada capa, etc.

4. Establecimiento de hiperparámetros. Se determinan los valores y tipos de hiperparámetros que vamos a utilizar.

5. Propagación hacia adelante (Forward Propagation). Se alimentan los datos de entrenamiento a través del modelo para obtener sus predicciones. Es la manera en la cual las redes neuronales crean las predicciones. En un principio la red neuronal tiene valores de W y b aleatorios en cada neurona. Los datos de entrenamiento pasan por estas neuronas hasta llegar a la capa de salida. En esta capa la red neuronal predice la clase a la cual pertenecen los datos de entrenamiento. Estas predicciones las usa la función de perdida para medir el grado de bondad de la red neurona.

Este ciclo se repite varias veces según indiquemos y en cada ciclo se ejecuta el algoritmo de Backpropagation para actualizar los valores de W y b.

6. Cálculo de Pérdida (Loss Calculation). Se calcula la pérdida (loss) entre las predicciones del modelo y las etiquetas verdaderas de los datos de entrenamiento. El valor de la pérdida es una medida del correcto funcionamiento de ese modelo.

7. Backpropagation. Se utiliza un algoritmo de optimización (como el descenso del gradiente) para ajustar los parámetros del modelo y reducir la pérdida. Este proceso implica calcular gradientes a través de la red neuronal en dirección opuesta al gradiente de la función de pérdida. El algoritmo de optimización es un hiperparámetro. Aprendemos de los errores.

El encargado de optimizar la función de perdida para mejorar las predicciones de una red neuronal. Este algoritmo se encarga de calcular las derivadas (o gradientes) de los parámetros W y b para saber cómo estos parámetros afectan al resultado de la función de perdida. Para ser más precisos, la optimización de una red neuronal se divide en dos partes:

- La primera es el algoritmo de Backpropagation. Este algoritmo se encarga de ver como los valores de W y de b afectan al resultado de la función de perdida.

- La segunda parte es el algoritmo de optimización (hiperparámetro Optimizador). Éste se encarga de optimizar la red neuronal y cambiar los valores de W y de b conforme pasan los ciclos (o Epochs). Existen diferentes algoritmos de optimización, unos son mejores que otros, aunque depende del tipo de problema se este resolviendo. La idea general de todos ellos es encontrar el mínimo global de la función de perdida. Lo que hace el algoritmo es descender por la función hasta llegar a este punto donde la función se encuentra optimizada. Esto se logra con ayuda de las derivadas que indican que camino se tiene que seguir.

8. Actualización de parámetros. Se actualizan los parámetros del modelo utilizando los gradientes calculados durante la Backpropagation. Una vez que hemos aprendido de los errores, cambiamos para que no vuelvan a suceder.

9. Iteración. Los pasos 5 a 8 se repiten tantas veces como hayamos establecido determinando el hiperparámetro Epochs. Hasta que el modelo converge o se alcanza un cierto criterio de detención.

10. Evaluación del modelo. Una vez entrenado el modelo, se evalúa su rendimiento real utilizando el conjunto de datos de prueba (Test Set). En este punto introducimos al modelo datos que NO HA VISTO NUNCA. Por decir así, es la hora de la verdad. Calcularemos métricas de rendimiento como la precisión, el recall, la F1-score, etc. Hay muchas métricas y muchos procedimientos para medir el rendimiento de un modelo.

Todo este proceso se repite iterativamente ajustando los hiperparámetros del modelo y optimizando su rendimiento hasta que se logre el rendimiento deseado.

Veremos el rendimiento del modelo, probaremos diferentes configuraciones y volveremos a entrenar. Ajustaremos los hiperparámetros y terminaremos obteniendo los mejores resultados posibles.

2.5. Algunos Hiperparámetros y Otros Conceptos

Veamos alguno de estos hiperparámetros:

- Cost Function. La función de coste trata de determinar el error entre el valor estimado y el valor real, con el fin de optimizar los parámetros del algoritmo Machine Learning, Deep Learning, etc.

- Optimizador. Es el algoritmo que se utiliza para ajustar los parámetros de un modelo de manera que se minimice una función de pérdida o error durante el proceso de entrenamiento.

Los optimizadores funcionan mediante la actualización iterativa de los parámetros del modelo en función de la magnitud del gradiente de la función de pérdida con respecto a esos parámetros. El gradiente indica la dirección y la magnitud del cambio más pronunciado en la función de pérdida, lo que ayuda al optimizador a determinar en qué dirección y cuánto ajustar los parámetros para mejorar el rendimiento del modelo.

Prácticamente todos los optimizadores se basan en el concepto de descenso de gradiente (Gradient Descent). Veamos en qué consiste Gradient Descent.

El objetivo de Gradient Descent es encontrar el mínimo de una función.

La metáfora que suele utilizarse para esto es imaginar que estamos de noche en medio de una montaña y que el objetivo es alcanzar el punto más bajo. Si nos han dejado en un punto aleatorio siempre podemos ver que, en algunas direcciones, el terreno sube (no nos conviene: el objetivo es bajar) y que en otras baja (sí nos conviene).

El gradiente es una medida de la pendiente en un punto dado.

Y, de entre las que bajan, podemos escoger, gracias al gradiente, aquella dirección en la que el descenso es más pronunciado. Así que damos un paso en dicha dirección.

Y volvemos a plantearnos la situación: desde el nuevo punto en el que estamos ¿cuál es la dirección en la que el terreno baja más rápidamente? Y volvemos a dar un paso en dicha dirección. Y repetimos el proceso hasta que lleguemos a un punto en el que todas las direcciones nos lleven hacia arriba, es decir, cuando hayamos llegado a un mínimo.

Imaginemos siempre que somos una personita bajando poco a poco por la función de pérdida. Como si fuéramos andando por la superficie de la gráfica buscando el punto más bajo de todos.

Veamos un ejemplo muy sencillo de la función de pérdida:

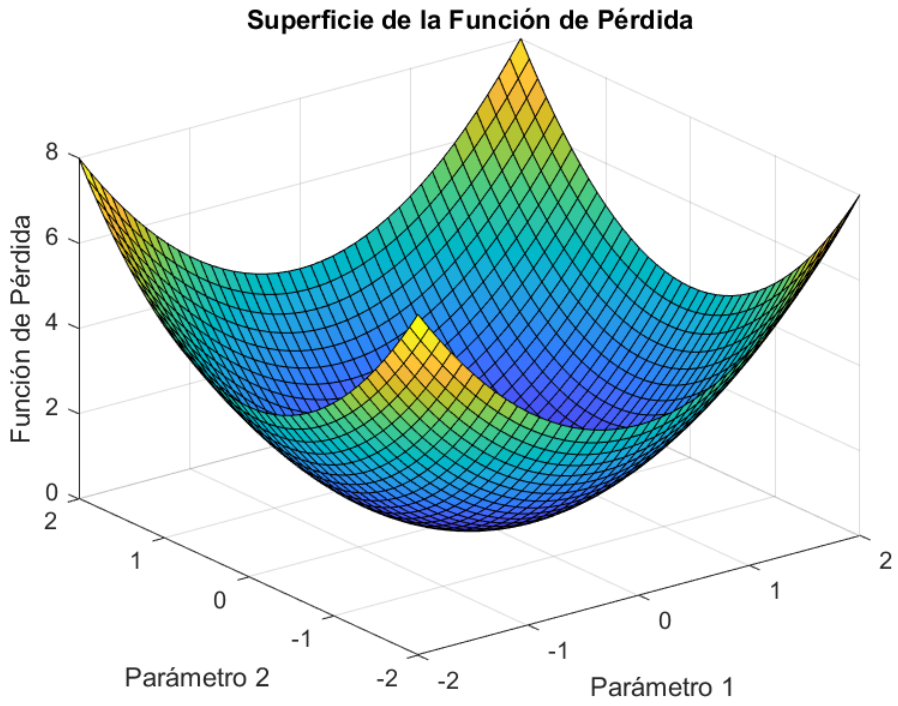


Figura 2.6: Ejemplo de Función de Pérdida

Ese ejemplo de función de pérdida es sencillo porque siempre es descendente hasta llegar al valle, al mínimo global. Una vez localizada la 'cuesta abajo', simplemente tenemos que seguirla. Esto sucede además desde cualquier punto.

Ahora supongamos una función de pérdida más complicada, con varios mínimos locales y un único mínimo total. Los mínimos locales son mínimos secundarios, de menor envergadura que el mínimo global, que es el mínimo principal, 'el más mínimo'.

Veamos la siguiente gráfica:

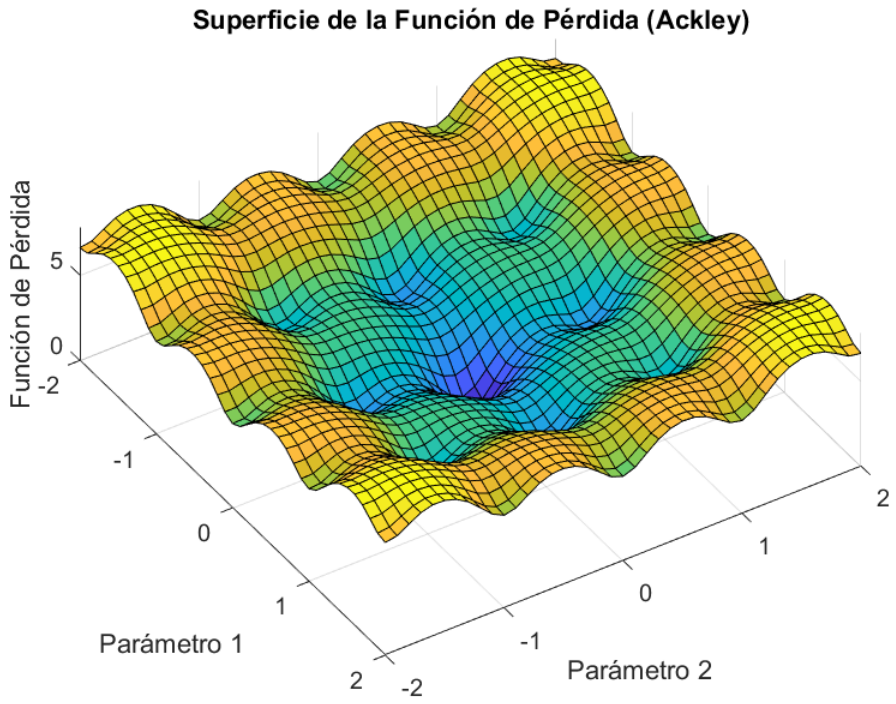


Figura 2.7: Ejemplo de Función de Pérdida (Función de Ackley)

La función de Ackley es una función de prueba fantástica, comúnmente utilizada en optimización numérica y en el análisis de algoritmos de optimización, como el descenso de gradiente. Fue propuesta por David Ackley en 1987[1]. La función de Ackley se define de la siguiente manera:

$$\begin{aligned} f(x,y) = &- 20 \times \exp\left(-0{,}2 \times \sqrt{0{,}5 \times (x^2 + y^2)}\right) \\ &- \exp\left(0{,}5 \times (\cos(2\pi \times x) + \cos(2\pi \times y))\right) + 20 + e \end{aligned} \tag{2.3}$$

Donde x y y son las variables de entrada.

Desde luego es una fórmula compleja. Analicemos la gráfica resultante. Esta función tiene las siguientes características:

1. Tiene un mínimo global en $f(0,0) = 0$, que es el mínimo absoluto de la función.

2. Tiene varios mínimos locales.

3. La función tiene una forma característica de cuenco con regiones estrechas y muchas oscilaciones.

4. La función puede ser difícil de optimizar debido a sus múltiples mínimos locales y a las oscilaciones en el paisaje de la función.

Veamos esos mínimos locales y el mínimo global:

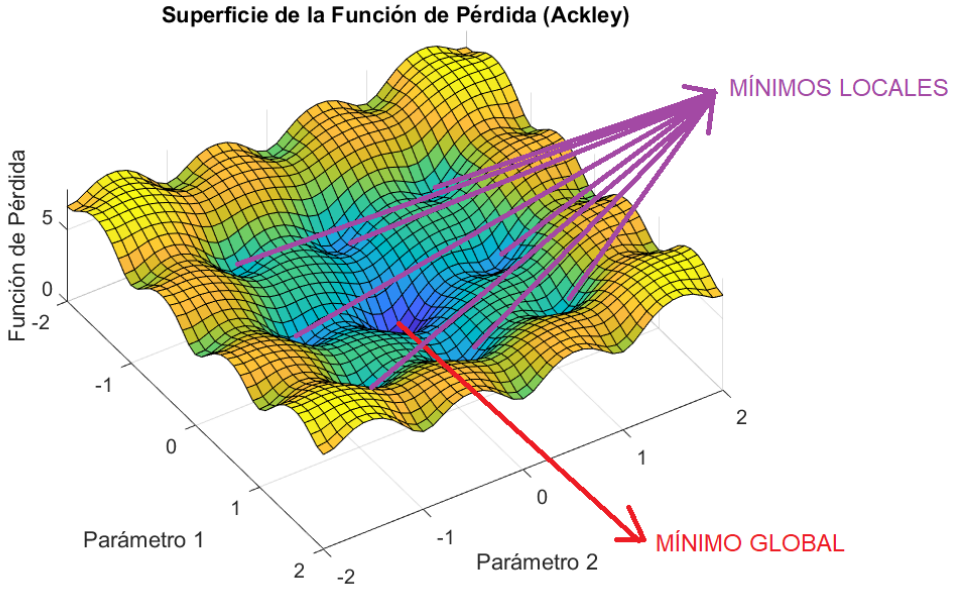


Figura 2.8: Mínimo Global y Mínimos Locales

Debido a estas características, la función de Ackley se utiliza comúnmente como una función de prueba para evaluar la eficacia y el rendimiento de los algoritmos de optimización en la búsqueda de mínimos globales en espacios de búsqueda multidimensionales.

Como se puede ver, es muy difícil conseguir detectar ese mínimo global. Supongamos que estamos andando por la gráfica, buscando ese mínimo global. Podemos llegar a un mínimo local y pensar que ese es el global. Y quedarnos ahí tan contentos, sin saber que hay otro mínimo más abajo, que sí es el global. Y estaremos equivocados.

Desde luego, es una gráfica fantástica.

Los datos de entrenamiento ayudan a que estos modelos aprendan con el tiempo, y la función de costo dentro del descenso de gradiente actúa específicamente como un barómetro, midiendo su precisión con cada iteración de actualizaciones de parámetros. Hasta que la función sea cercana o igual a cero, el modelo continuará ajustando sus parámetros para producir la menor cantidad de errores posible.

El objetivo del descenso de gradiente es minimizar la Cost Function, o el error entre la predicción del modelo y la realidad. Para hacer esto, se necesitan dos puntos de datos: una dirección y una tasa de aprendizaje. Estos factores determinan los cálculos de derivadas parciales de iteraciones futuras, lo que le permite llegar gradualmente al mínimo local o global (es decir, punto de convergencia).

- Learning Rate (tamaño de paso o alfa). Es el tamaño de los pasos que se dan para alcanzar el mínimo. La envergadura de los pasitos que damos mientras caminamos por la gráfica buscando ese mínimo global. Suele ser un valor pequeño y se evalúa y actualiza en función del comportamiento de la función de costos. Si su valor es elevado, las tasas de aprendizaje dan como resultado pasos más grandes, pero se corre el riesgo de sobrepasar el mínimo. Por el contrario, una tasa de aprendizaje baja tiene tamaños de paso pequeños. Si bien tiene la ventaja de una mayor precisión, el número de iteraciones compromete la eficiencia general, ya que esto requiere más tiempo y cálculos para alcanzar el mínimo.

La Cost Function (o función de pérdida) mide la diferencia, o error, entre la y real y la y pronosticada en su posición actual. Esto mejora la eficacia del modelo de Machine Learning, proporcionando feedback al modelo para que pueda ajustar los parámetros para minimizar el error y encontrar el mínimo local o global. Repite continuamente, moviéndose a lo largo de la dirección de descenso más pronunciado (o el gradiente negativo) hasta que la función de costo está cerca o en cero. En este punto, el modelo dejará de aprender. Además, si bien los términos función de costo y función de pérdida se consideran sinónimos, existe una ligera diferencia entre ellos.

Una función de pérdida se refiere al error de un ejemplo de entrenamiento, mientras que una función de costo calcula el error promedio en todo un conjunto de entrenamiento.

- Tamaño del Lote (Batch Size). Es el número de muestras de datos que se utilizan para actualizar los pesos del modelo en cada iteración del algoritmo de optimización. Un tamaño de lote más grande puede acelerar el proceso de entrenamiento, pero puede requerir más memoria. Un tamaño de lote más pequeño

puede proporcionar una convergencia más suave y puede ser útil para modelos con conjuntos de datos grandes.

- Número de Épocas (Epochs). Representa el número de veces que el algoritmo de entrenamiento pasa por todo el conjunto de datos durante el proceso de entrenamiento. Entrenar durante más épocas puede mejorar la precisión del modelo hasta cierto punto, pero también puede conducir al sobreajuste si se entrena demasiado.

- Función de Activación. Determina la salida de una neurona y su impacto en la red neuronal. Ejemplos comunes incluyen la función de activación sigmoide, ReLU (Rectified Linear Unit), tanh (tangente hiperbólica), entre otras.

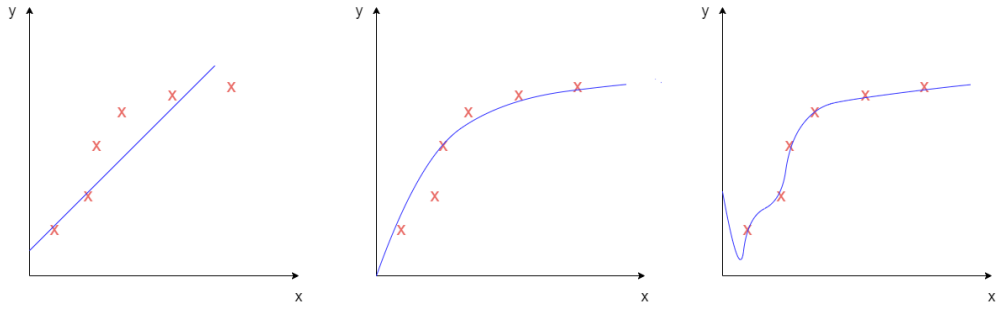
- Regularización. Ayuda a prevenir el sobreajuste al agregar términos adicionales a la función de pérdida que penalizan la complejidad del modelo. Ejemplos de técnicas de regularización incluyen la regularización L1, L2, dropout, entre otras.

2.6. La Importancia de un Buen Entrenamiento

Podemos pensar que, cuanto más entrene el modelo con esos datos, mejor. Esto es incorrecto por el problema denominado Overfitting.

Overfitting: Si el modelo trabaja con demasiadas características (features), podemos obtener un modelo que funciona excelentemente para ese conjunto de datos de entrenamiento, está perfectamente ajustado a esos datos, pero su rendimiento será malo a la hora de generalizar ante datos nuevos.

Otro problema es el del **Underfitting**, donde el modelo ha aprendido poco, puede aprender más. El resultado, por ejemplo, es una curva de extrapolación muy simplista.



(a) Underfitting **(b)** Entrenamiento Óptimo **(c)** Overfitting

Figura 2.9: Diferencia entre Underfitting, Overfitting y Entrenamiento Óptimo

En el ejemplo, la fórmula de la curva correspondiente al caso de Underfitting (línea recta) es:

$$y = \Theta_0 + \Theta_1 \cdot x \tag{2.4}$$

La curva de entrenamiento óptima corresponde a una gráfica de segundo grado cuya ecuación es:

$$y = \Theta_0 + \Theta_1 \cdot x + \Theta_2 \cdot x^2 \tag{2.5}$$

En el caso de Overfitting, la curva pasa por todos los puntos de una manera un tanto forzada. La fórmula de esta curva sería similar a:

$$y = \Theta_0 + \Theta_1 \cdot x + \Theta_2 \cdot x^2 + \Theta_3 \cdot x^3 + \Theta_4 \cdot x^4 \tag{2.6}$$

Por lo tanto, no debemos buscar una curva muy simplista ni una curva artificialmente ajustada. Buscaremos una curva que se ajuste suficientemente a los puntos.

2.7. Procedimiento para el Entrenamiento

El primer paso es organizar el dataset. El dataset es el conjunto de datos con el que vamos a trabajar.

Una vez que tenemos todo ese conjunto de datos, el dataset, lo dividimos en tres bloques:

1. Conjunto de entrenamiento (Training Set).

Es el conjunto de datos utilizado para entrenar el modelo. Contiene ejemplos de entrada y las respuestas correctas (etiquetas o salidas esperadas).

Durante el proceso de entrenamiento, el modelo ajusta sus parámetros para minimizar la discrepancia entre las salidas predichas por el modelo y las salidas reales proporcionadas en el conjunto de entrenamiento.

Insistimos: Es fundamental un Training Set completo, representativo y que contenga todas las casuísticas. De ello va a depender la capacidad del modelo para funcionar adecuadamente ante datos que no haya visto nunca.

2. Conjunto de validación (Validation Set).

Es un conjunto de datos utilizado para ajustar los hiperparámetros del modelo y evaluar su rendimiento durante el entrenamiento.

A diferencia del Training Set, el Validation Set no se utiliza para ajustar los parámetros del modelo directamente. Se utiliza para evaluar el rendimiento del modelo en datos no vistos durante el proceso de entrenamiento y así ajustar los hiperparámetros. Como ven, el ajuste de hiperparámetros es un proceso relativamente manual y, frecuentemente, sujeto a ensayo y error. Vas probando, compruebas los resultados obtenidos en el Validation Set y compruebas si el modelo funciona mejor o peor con este u otro valor del hiperparámetro.

A medida que el entrenador adquiere experiencia, este proceso manual y empírico se acelera porque ese entrenador ya tiene 'olfato'.

Por cierto, los resultados en el Validation Set ayudan a evitar el Overfitting del modelo ya que proporciona una estimación independiente del rendimiento del modelo.

3. Conjunto de prueba (Test Set).

Es un conjunto de datos independiente utilizado para evaluar el rendimiento final del modelo después de que se haya entrenado y ajustado completamente.

Ya hemos terminado de entrenar el modelo y ahora llega el momento de comprobar cómo funciona ante datos que no ha visto nunca. Es la hora de la verdad.

El Test Set se utiliza una vez que el modelo ha sido entrenado y se ha seleccionado el mejor conjunto de hiperparámetros basado en el rendimiento en el conjunto de validación.

Proporciona una evaluación imparcial del rendimiento del modelo en datos completamente nuevos y no vistos durante el entrenamiento o la validación.

Es fundamental que el modelo NUNCA HAYA VISTO DE NINGUNA MANERA POSIBLE los datos del Test Set. Por supuesto, jamás se deben utilizar datos del Training o Validation Sets.

Por lo tanto y a modo de resumen, las 3 etapas de un entrenamiento:

- Training: Entrenamos el modelo para que aprenda de los datos.

- Validation: Validamos el modelo. Ajustamos y evaluamos los hiperparámetros del modelo.

- Test: Comprobamos el rendimiento del modelo con datos nunca vistos por el modelo.

Un buen reparto del dataset entre estos tres grupos es:

- Training Set: 60 %

- Validation Set: 20 %

- Test Set: 20 %

Esta configuración es comúnmente aceptada. Recomendamos que todos los datos nuevos que puedan surgir, sean Test Set para verificar adicionalmente que el modelo funciona bien.

Un consejo adicional. Si los datos están desbalanceados y no se pueden balancear porque en la vida real esta es la situación, recomendamos modificar el balance en el Test Set.

Supongamos dos categorías excluyentes, es decir, un elemento o es A o es B. No hay más opciones. En la vida real tenemos un 70 % de datos de una categoría A y sólo en 30 % de la categoría B (es decir, 'no A'). Es decir:

- 700.000 datos de A.

- 300.000 datos de B.

Podemos trabajar con 300.000 datos de A y 300.000 datos de B. Pero es una pena perder la información de los 400.000 datos de A que no vamos a tener en cuenta.

En estos casos hay una prueba que merece la pena hacer. Mantenemos este desbalance 70 %/30 % en el Training y Validation Sets. Y en el Test Set, trabajamos con un 50 %/50 %. De esta manera, en el Test Set desaparece ese sesgo por cantidad entre ambas categorías.

2.8. ¿Cómo Detectar un Buen Entrenamiento?

Indudablemente, un modelo que se ha entrenado correctamente es aquel que funciona correctamente ante CUALQUIER dato de entrada relacionado con su entrenamiento.

Evidentemente, es muy poco operativo comprobar que un modelo funciona introduciendo miles, millones de datos de entrada que no haya visto nunca, esperar que genere datos de salida, procesarlos e interpretarlos.

Hay ciertas dinámicas que pueden ayudar. Veamos la siguiente gráfica:

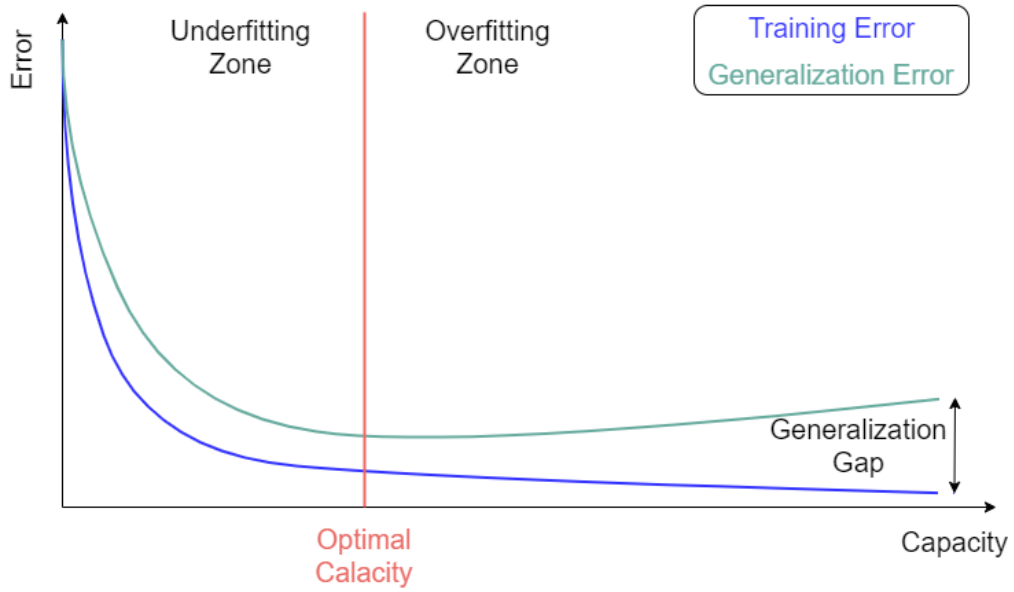


Figura 2.10: Desarrollo de la Curva de Entrenamiento

Ante una curva de entrenamiento, cuando la pendiente descendiente de la curva de error es pronunciada, aún estamos en Zona Underfitting (poco entrenamiento). Así que queda margen de mejora.

Llegada a la región de meseta (más o menos), en el momento en el cual la curva de generalización asciende mientras que la Training desciende, en ese momento hemos entrado en la Zona Overfitting (exceso de entrenamiento).

Es decir, necesitamos encontrar el punto intermedio entre estas dos zonas.

Recomendamos jugar con el número de Epochs. Excedernos en el número de Epochs, visualizar el Overfitting y, ahí, tomar la medida aproximada del momento en el que empieza el Overfitting. Y ese será el punto óptimo aproximado.

Veamos una serie de ejemplos prácticos, casos reales. Las siguientes curvas de entrenamiento se generaron entrenando modelos utilizando MATLAB.

En este caso primer caso tenemos un claro caso de Overfitting.

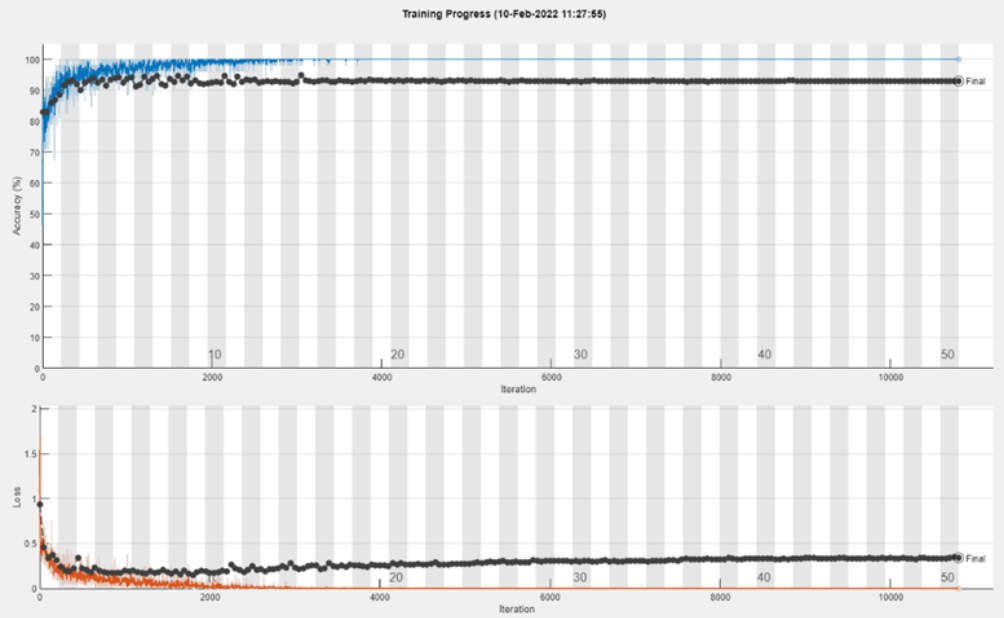


Figura 2.11: Ejemplo de Overfitting.

Las curvas azul y naranja representan el grado de acierto del modelo en el Training Set. La azul es el grado de acierto (accuracy, precisión) y la naranja el grado de error (loss, pérdida). Las curvas negras, ídem en Validation Set. Cuando sucede overfitting, vemos que el modelo alcanza un grado de acierto del 100 %. La curva azul se estabiliza completamente en el tope máximo del eje y. Lo mismo con la curva naranja, se estanca en el 0 %. Un grado de acierto del 100 % en el Training Set suele ser una mala señal. Especial atención si esto sucede.

Mientras tanto, la curva Loss en el Validation Set (parte inferior, curva negra) comienza a aumentar. Es decir, la función de pérdida degenera en el Validation Set.

Estos son los dos indicios visuales que nos llevan a detectar el Overfitting. Si aplicamos a este modelo un Test Set, veremos que el índice de error es grande. El modelo no es capaz de generalizar, no ha aprendido. Dicho de una manera burda, 'el modelo se ha aprendido las respuestas del Training Set'.

Algunos consejos para solucionar un problema de Overfitting o solucionarlo:

- Entrenar con más datos, con datos nuevos (si es posible).

- Data Augmentation: aplicar modificaciones sobre los datos existentes para obtener 'datos nuevos' (rotar imágenes, ampliar, reducir, voltear simétricamente...).

- Añadir ruido a los datos de entrada.

- Feature selection. Eliminar características redundantes. Imaginemos que una feature es x y otra feature es x^2. Son variables totalmente correladas y podemos simplificar x^2 a x. Por lo tanto, podemos prescindir de x^2 realmente.

- Cross-validation. Es una medida robusta para evitar el Overfitting. El conjunto de datos contenido en los Training y Validation Sets se divide en partes. En la validación cruzada estándar de k pliegues (k-folds), tenemos que dividir los datos en k pliegues. A continuación, entrenamos iterativamente el algoritmo en k-1 pliegues mientras utilizamos el pliegue restante como conjunto de prueba. Este método nos permite ajustar los hiperparámetros de la red neuronal o del modelo de aprendizaje automático y probarlo con datos completamente desconocidos.

- Simplificar. Disminuir la complejidad del modelo. Por ejemplo, 'podar' un árbol de decisión, reducir el nº de parámetros en una red neuronal o utilizar dropout (desactivar un numero de neuronas de una red neuronal de forma aleatoria).

- Regularización. Si se da Overfitting en un modelo muy complejo, podemos reducir el valor de los parámetros del modelo. La regularización consiste en penalizar de alguna forma las predicciones que hace nuestra red durante el entrenamiento, de forma que 'no piense que el Training Set es la verdad absoluta' y así pueda generalizar mejor cuando ve otros datasets. Algunos métodos: L2 (Lasso regularization), L1 (Ridge regularization), regularización por restricción (Max norm constraints).

- Ensembling. Es una técnica de aprendizaje automático que combina varios modelos base para producir un modelo predictivo óptimo. En el Ensemble Learning, las predicciones se agregan para identificar el resultado más popular. Entre los métodos más usuales tenemos bagging (algoritmos simples en paralelo) y boosting (algoritmos simples en serie, secuencialmente).

- Early stopping. Este método pretende detener el entrenamiento del modelo antes de memorizar el ruido y las fluctuaciones aleatorias de los datos. Puede existir el riesgo de que el modelo deje de entrenar demasiado pronto, lo que llevaría a un ajuste insuficiente. Hay que llegar a un tiempo/iteraciones óptimos para que el modelo se entrene.

- Añadir capas dropout. Los pesos grandes en una red neuronal significan una red más compleja. La eliminación probabilística de nodos de la red es un método sencillo y eficaz para evitar el Overfitting. En la regularización, algunas salidas de las capas se ignoran o 'descartan' aleatoriamente para reducir la complejidad del modelo.

IMPORTANTE: es importante reducir la complejidad del modelo.

Si tenemos dos modelos con un rendimiento casi igual, con la única diferencia de que uno de ellos es más complejo que el otro, siempre se debe optar por el modelo menos complejo.

Regla fundamental en ciencia de datos: empezar siempre con un modelo menos complejo e ir añadiendo complejidad gradualmente.

El otro problema habitual es el Underfitting. Lo contrario a Overfitting. La red neuronal no ha aprendido correctamente sobre los datos de entrenamiento y en general tiene un desempeño muy pobre en todas las predicciones.

Sus posibles motivos:

- No hay suficientes parámetros o complejidad para modelar adecuadamente los datos.

- Los priores bayesianos son demasiado restrictivos o ciertos (baja entropía). Los priores son las distribuciones de probabilidad que expresan nuestras creencias o conocimientos previos sobre los parámetros de un modelo antes de observar los datos. Si los priores son demasiado específicos o estrechos, asignan una alta probabilidad

a un conjunto limitado de valores de parámetros. Esto puede limitar la capacidad del modelo para capturar la variabilidad o incertidumbre en los datos y puede llevar a conclusiones excesivamente confiadas o sesgadas. Si los priores tienen baja entropía, significa que son muy específicos y asignan una probabilidad cercana a 1 a un conjunto muy limitado de valores de parámetros. En este caso, los priores están imponiendo fuertes restricciones en el modelo, lo que puede llevar a una inferencia excesivamente determinista o inflexible.

- No se le dio suficiente tiempo al algoritmo de aprendizaje automático para entrenar.

- Escasos datos de entrenamiento. Nuestro modelo no será capaz de generalizar el conocimiento.

Supongamos un modelo al que le enseñamos sólo una raza de perros y pretendemos que pueda reconocer a otras 10 razas de perros distintas. El algoritmo no será capaz de darnos un resultado bueno por falta de 'materia prima' para hacer sólido su conocimiento. También es ejemplo de Underfitting cuando la máquina reconoce todo lo que 've' como un perro, tanto una foto de un gato o un coche.

En el caso de underfitting, las curvas son simplemente pobres, no hay un grado de acierto adecuado ni en el Training Set ni en el Validation Set.

Hay casos de Underfitting muy interesantes. Ya hemos visto que tenemos que buscar un mínimo global en la curva de coste y ese será nuestro punto óptimo de funcionamiento. Si tenemos dos mínimos, el global y uno local (un mínimo en una zona pero que es peor que el global) y nuestro entrenamiento se atasca en ese mínimo local, sucede esto:

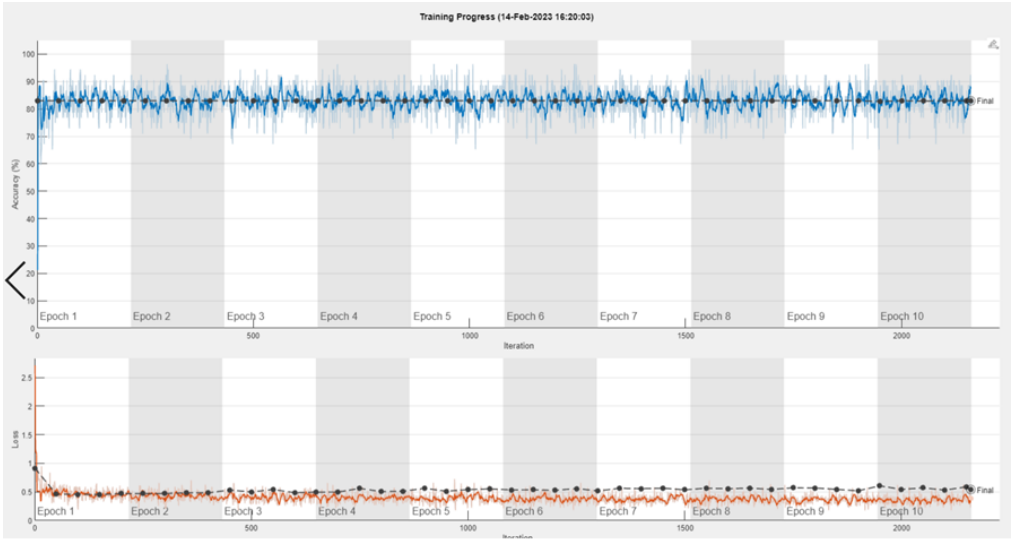


Figura 2.12: Ejemplo de Underfitting

En este caso el modelo clasifica entre dos categorías desbalanceadas (la Categoría 1 tiene un 80 % del total de datos del Training y Validation Sets y la categoría 2 el 20 % restante). El modelo simplemente clasifica todo como Categoría 1, así consigue, sin esfuerzo, una accuracy del 80 %. Es una buena nota y, si este modelo fuera humano y estudiante, desde luego sería una estrategia muy inteligente para superar una asignatura.

Pero no es humano. Así que lo único que tenemos es un modelo que no ha aprendido nada. La solución de este caso de Underfitting es compleja y requiere un rediseño del dataset para eliminar ese atasco en el mínimo local. Veamos el motivo.

Para finalizar, vamos a ver un caso real con una buena curva de entrenamiento.

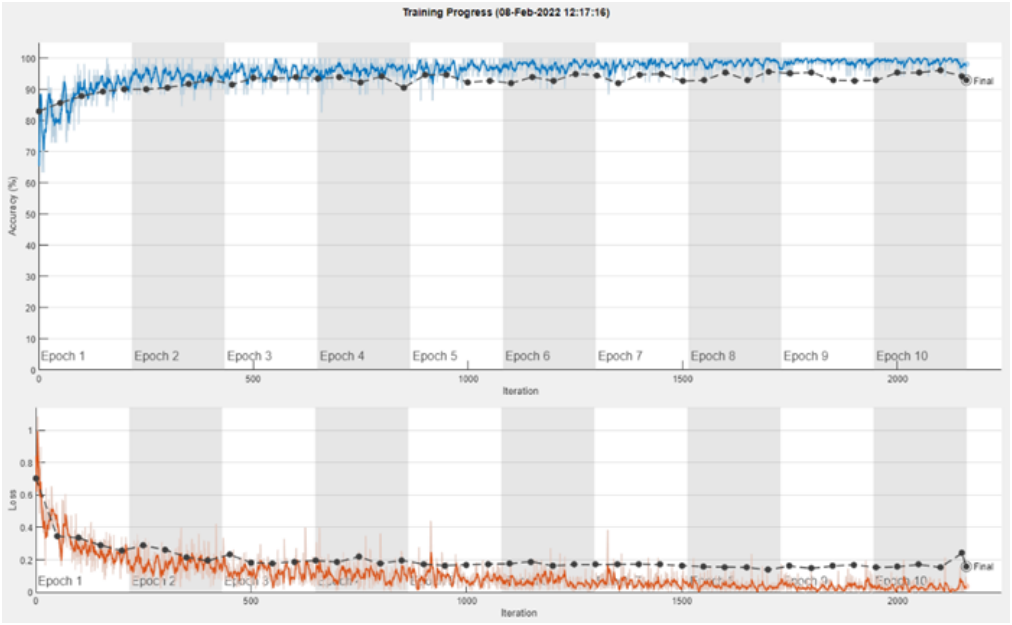


Figura 2.13: Ejemplo de un Buen Entrenamiento.

¿Por qué es un buen entrenamiento? Porque a simple vista podemos comprobar que la red no ha alcanzado un grado de acierto (accuracy) del 100 %. Se queda en un 98 % aprox. en el Training Set y en un 93 % aprox. en el Validation Set. Esto es bueno, denota que la red tiene capacidad para generalizar. Esta red neuronal consiguió un acierto del 94 % en el Test Set, con porcentajes de acierto similares en todas las categorías a clasificar.

Recordamos que estos modelos tienen mucha estadística y probabilidad en su funcionamiento para aprovechar esa capacidad de generalización. Esto implica necesariamente asumir un porcentaje de error. Si este error es lo suficientemente pequeño para la función del modelo, perfecto.

En un buen entrenamiento, sin anomalías, las curvas correspondientes a Training Set deben ser ligeramente mejores que las correspondientes al Validation Set. Si esto no sucede, tenemos que investigar bien el motivo porque puede querer decir que hay un problema en el entrenamiento.

Todo lo expuesto en este apartado nos da pistas acerca de si el entrenamiento ha salido bien o mal, si hay Overfitting o Underfitting.

Pero la prueba de fuego es el Test Set. Un modelo funciona bien cuando acierta de una manera solvente en datos que no ha visto nunca. Pero no sólo. Tiene que acertar de una manera solvente FUNCIONANDO ADECUADAMENTE. Aquí empieza el problema.

Tener una buena Test Accuracy (precisión, grado de acierto en el Test Set) es bueno, pero no es suficiente. Tenemos que asegurarnos de que detrás de esos aciertos hay una detección correcta de patrones.

Me explico utilizando de nuevo un clasificador de imágenes.

Supongamos que tenemos un modelo que detecta perfectamente motos de nieve. Es capaz de detectar en cualquier imagen una moto de nieve. Las detecta estando en cualquier posición, a cualquier distancia, con muy bajas resoluciones... Es un gran detector de motos de nieve.

Hasta que un día prueban el modelo con una moto de nieve en un remolque dentro de un garaje. La moto de nieve es perfectamente visible, grande, nítida, centrada en medio de la foto. Pero el modelo no la detecta. ¿Cómo es posible?

Porque en el suelo no hay nieve. Todas las fotos con las que el modelo entrenó consistían en motos de nieve en un entorno nevado, con el suelo lleno de nieve. La red sólo detecta motos de nieve si el suelo es blanco.

Rescataremos este ejemplo a lo largo de este libro porque es muy representativo de las problemáticas inherentes a los modelos Machine Learning y Deep Learning.

Por lo tanto, insistimos, aunque tengamos muy buenos resultados en cuanto a grado de acierto en el Test Set (Test Accuracy), aún tenemos que hacer muchas comprobaciones adicionales antes de empezar a creernos que el modelo funciona.

2.9. Técnicas de Comprobación del Funcionamiento del Modelo

Hay muchas maneras de realizar comprobaciones adicionales acerca del grado de acierto real de un modelo. Nos limitamos a citar las que más nos gustan por su sencillez y buen rendimiento.

2.9.1. Curva ROC y AUC (Area Under Curve)

Una curva ROC (curva de característica operativa del receptor) es un gráfico que muestra el rendimiento de un modelo de clasificación en todos los umbrales de clasificación.

En el eje x (horizontal) se representa la Tasa de Falsos Positivos (False Positive Rate, FPR), que es la proporción de casos negativos que son incorrectamente clasificados como positivos. En el eje y (vertical) se representa la Tasa de Verdaderos Positivos (True Positive Rate, TPR), que es la proporción de casos positivos que son correctamente clasificados como positivos.

Veamos sus fórmulas:

$$\text{FPR} = \frac{\text{Falsos Positivos}}{\text{Falsos Positivos} + \text{Verdaderos Negativos}} \tag{2.7}$$

$$\text{TPR} = \frac{\text{Verdaderos Positivos}}{\text{Verdaderos Positivos} + \text{Falsos Negativos}} \tag{2.8}$$

El AUC mide el área bidimensional completa debajo de la curva ROC completa de (0,0) a (1,1).

La siguiente figura representa el esquema de AUC y la diferencia entre buenos valores de AUC y valores deficientes.

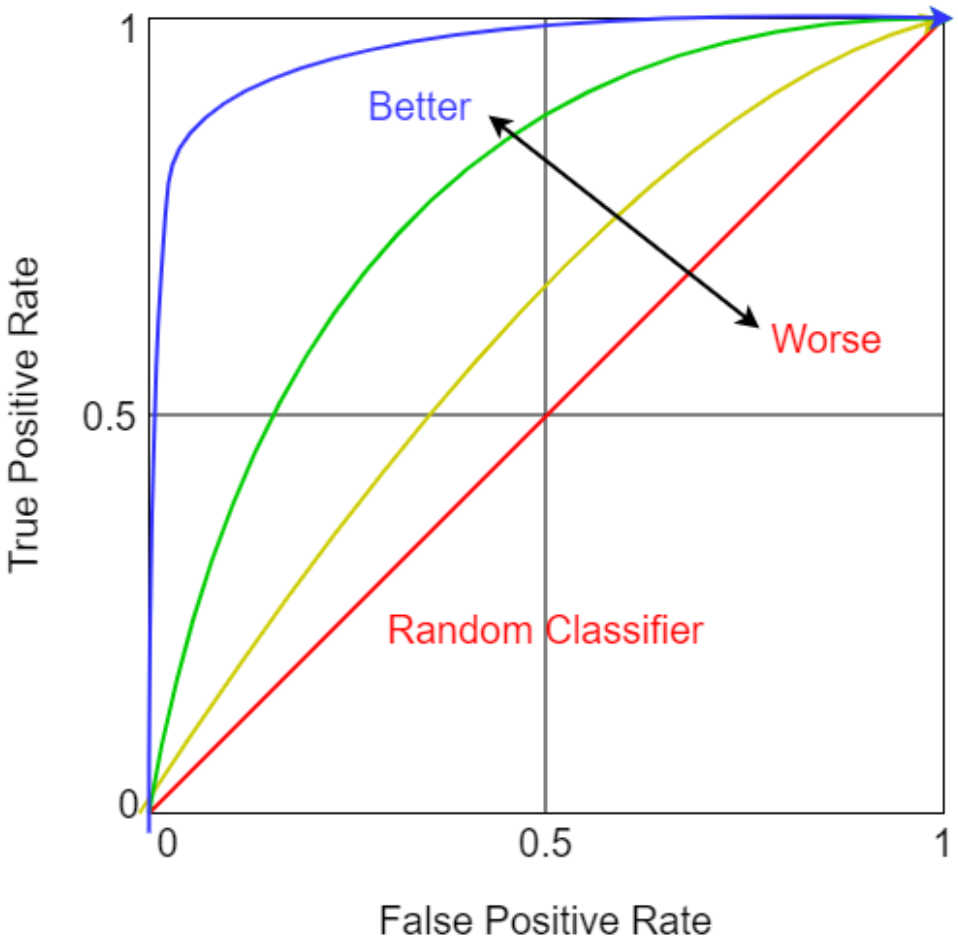


Figura 2.14: Esquema de AUC

2.9.2. Confusion Matrix

Otra medida de un buen funcionamiento de un modelo es la Confusion Matrix. Es una medida excelente.

Permite visualizar el grado de acierto de un algoritmo de clasificación por categorías. Nos da información no sólo del rendimiento del algoritmo sino del grado de equilibrio entre las categorías.

Si tenemos un clasificador de perros y gatos que funciona excelentemente con perros y relativamente bien con gatos, tenemos un modelo con capacidad de mejora (en gatos).

El esquema de una Confusion Matrix es el siguiente:

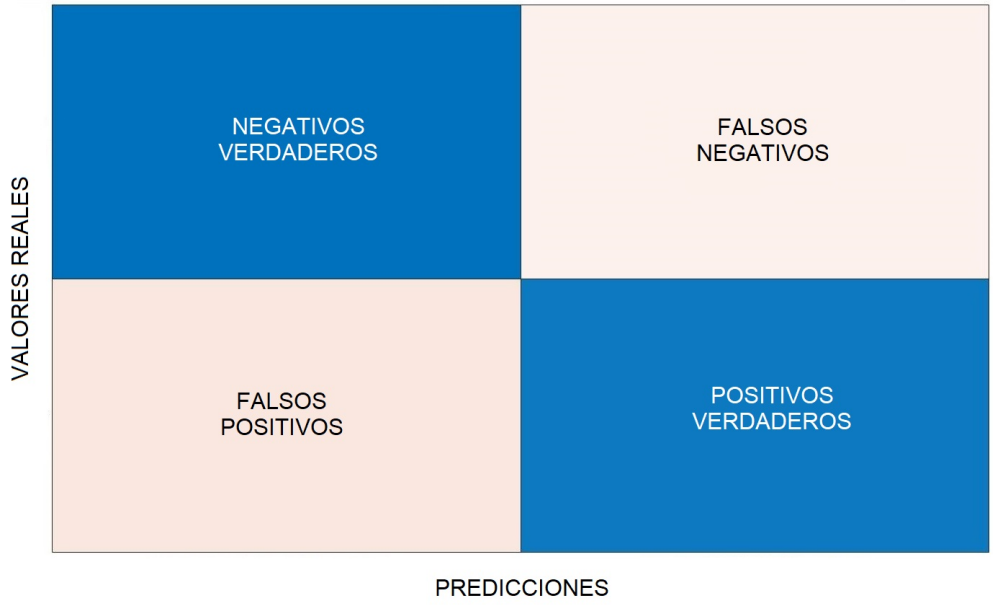


Figura 2.15: Esquema de Confusion Matrix

Pero a continuación vamos a ver la mejor manera de comprobar un buen funcionamiento de un modelo: las herramientas de interpretabilidad.

2.9.3. Herramientas de Interpretabilidad

Las redes neuronales tienen fama de 'cajas negras'. No sabemos cómo funcionan. No sabemos por qué toman sus decisiones.

Son cajas negras.

No obstante, se han desarrollado herramientas para, caso de imágenes, saber DÓNDE mira la red para tomar una decisión. No sabemos qué ve pero sabemos dónde mira.

Estas herramientas son útiles para comprobar el correcto funcionamiento de la red más allá de la mera cuantificación de tasas de acierto, medida del AUC, Confusion Matrix, etc.

Estas herramientas son muy útiles no sólo para detectar si estas redes realmente funcionan mal, sino incluso para detectar sesgos.

Las zonas donde mira la red para decidir deben ser coherentes con la realidad. Si una red decide que una imagen de un perro es un perro pero mira el comedero

de a lado con una imagen de un hueso: funciona mal independientemente del grado de acierto de la misma. Si mira nariz, orejas, etc. será correcto.

Vamos a ver un ejemplo de un funcionamiento correcto que hemos podido comprobar gracias a la herramienta de interpretabilidad Grad-CAM. Es una herramienta de interpretabilidad muy intuitiva y visual, además, bajo requerimiento computacional. Es poco precisa en ocasiones pero funciona francamente bien.

Sobre la imagen original, aplica una escala de color para remarcar las zonas relevantes para la decisión del modelo. En este caso, la escala de color aplicada para realzar las zonas importantes es la denominada Jetmap. En esta escala, la importancia se gradúa desde el azul oscuro (poco o nada importante) hasta el rojo intenso (máxima importancia) con la siguiente transición de colores:

Figura 2.16: Escala de Color Jetmap

Supongamos un modelo que clasifica perros y gatos. Introducimos en el modelo una foto de un perro y la clasifica correctamente. Aplicamos Grad-CAM y obtenemos las zonas de la imagen que, según ese modelo, han sido determinantes para su clasificación como perro:

(a) Imagen Original **(b)** Imagen Grad-CAM

Figura 2.17: Aplicación Grad-CAM a la Clasificación de un Perro. Fotografía suministrada por Izaskun Kintana Arkoitza.

Para determinar que esa imagen es de un perro, ha mirado sobre todo la zona de las orejas, las cercanías a la nariz y, en menor medida, el lomo y las patas.

Por lo tanto, para decidir que es un perro, ha mirado las zonas lógicas para tomar esa decisión.

Veamos ahora cómo clasifica un gato:

(a) Imagen Original **(b)** Imagen Grad-CAM

Figura 2.18: Aplicación Grad-CAM a la Clasificación de un Gato. Fotografía suministrada por Fernando José Sadio-Ramos.

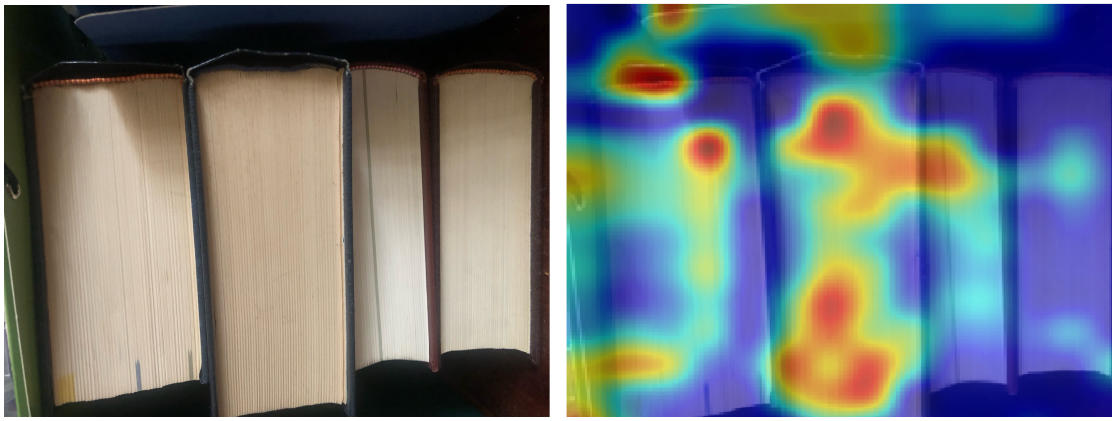
Para determinar que es un gato, la red se ha fijado en la parte superior de los ojos, el morro, las orejas y los mofletes. Tiene sentido.

Por lo tanto, la red ha mirado las zonas lógicas para decidir que es un gato y así lo ha decidido.

Entendemos que el funcionamiento de la red es correcto, al menos parece coherente.

Por último, vamos a ver un caso de funcionamiento controvertido y aplicamos interpretabilidad.

Vamos a introducir al modelo una imagen de difícil interpretación, tal vez incluso para el ojo humano. Vamos a analizar qué sucede:



(a) Imagen Original **(b)** Imagen Grad-CAM

Figura 2.19: Aplicación Grad-CAM a la Clasificación de Libros Apilados. Fotografía suministrada por Ana Guerrero Tamayo.

En este caso, el modelo identificó estos libros apilados con un armario ropero. Es decir, cometió un error. En su favor, diremos que rotar la imagen hace que sea más confuso determinar que son libros apilados (es un desafío a la Ley de la Gravedad).

Al aplicar interpretabilidad, podemos deducir que la red no tiene muy claro dónde mirar.Hay muchas zonas relevantes, muy desperdigadas y no está muy claro que esas zonas sigan un patrón concreto. Las zonas detectadas por Grad-CAM no parecen muy coherentes. Resulta todo muy confuso.

Podríamos pensar (cuidado, sólo podríamos pensar) que el modelo no ha tenido claro dónde mirar para detectar algún patrón que coincida con alguna de las categorías que conoce.

En definitiva, esto es lo que nos ofrece la interpretabilidad: la posibilidad de ver dónde mira la red para decidir.

Pero, insistimos, no sabemos qué ve.

Sólo sabemos dónde mira.

A fecha de marzo de 2024, aún queda mucho por hacer en cuanto a conocer lo realmente importante: qué ve el modelo para tomar sus decisiones.

La máquina tecnológicamente más eficiente que el hombre ha inventado es el libro.

— Northrop Frye, *filósofo, lingüista y teólogo. Doctor 'honoris causa' por una treintena de universidades europeas y americanas. Decía que no podía tocar en público el piano porque un mono balbuceante le decía a intervalos 'bien, ahora es hora de que cometas un error'.*

3 Algunas arquitecturas

Índice

3.1. Introducción

Como hemos indicado en el Capítulo 1, no vamos a utilizar el término IA. Hablaremos de Machine Learning, de Deep Learning. Y sobre todo, de modelos generativos porque esta denominación es más rigurosa.

La redacción de este libro finalizó en 2024, por lo que dentro de un tiempo probablemente breve, muchos de estos modelos ya habrán sido superados. No obstante, los principios de funcionamiento de los mismos pueden ayudar a comprender mejoras futuras que ya están en camino.

3.2. Convolutional Neural Network (CNN)

Red Neuronal Convolucional.

Es un tipo de red neuronal no generativa.

Estas redes son especialmente efectivas para tareas de visión por computadora, como clasificación de imágenes, detección de objetos y segmentación semántica. Si tenemos un dataset con imágenes, es una opción de modelo fantástica. Como tal, no son modelos generativos.

Aunque también se utiliza en análisis semántico.

Funciona excelentemente cuando los datos tienen forma matricial o se pueden convertir en matrices de números.

Son arquitecturas ya maduras, que han demostrado a lo largo del tiempo su potencia, sobre todo en datasets de imágenes.

Son excelentes, versátiles, y muy potentes en la detección de patrones no visibles por el ojo humano.

Además, aprovechan todas las ventajas del Transfer Learning. Es muy sencillo aplicar Transfer Learning sobre un modelo pre-entrenado para que aprenda de una nueva tarea con menor cantidad de datos.

Una CNN se compone fundamentalmente de varios tipos de capas diferentes:

- Capas convolucionales.

La convolución es la operación fundamental en las CNN. Consiste en deslizar un pequeño filtro (o kernel) sobre la matriz de entrada para extraer características locales. Estas características pueden ser bordes, texturas o patrones más complejos, dependiendo de la profundidad de la red. Ese kernel suele ser de 3x3 o de 5x5.

El kernel se desliza sobre la matriz de entrada mediante una operación llamada convolución. Durante este proceso, el kernel se coloca sobre un fragmento de la matriz de entrada y se multiplica elemento por elemento con los valores de píxeles correspondientes en ese fragmento.

Para cada posición del kernel en la imagen de entrada, se realiza un producto punto entre los valores de los píxeles del fragmento de la matriz de entrada y los

valores del kernel. Estos productos se suman para producir un único valor en una nueva matriz llamada Mapa de Características (Feature Map).

Y pasa al siguiente fragmento. Y así sucesivamente hasta que el kernel ha pasado por toda la matriz de entrada.

Este Feature Map tiene todos los resultados de ese paso del kernel por la matriz de entrada. Así que es compendio de las características capturadas por el kernel en diferentes partes de la matriz de entrada, es decir, de la imagen.

- Capas de Agrupación (Pooling Layers).

Sin querer entrar en muchos detalles técnicos, la capa Pooling comprime, por decir así, cada Feature Map generado por las capas convolucionales. Se divide cada mapa de características en regiones no superpuestas (por ejemplo, 2x2 o 3x3) y se aplica una operación de agrupación a cada región.

La operación de agrupación toma una región de entrada (por ejemplo de tamaño 2x2 o 3x3) y produce un solo valor de salida para esa región.

La operación más comúnmente utilizada es la operación de máximo (Max Pooling), que simplemente toma el valor máximo de la región de entrada. Otras operaciones comunes incluyen el promedio (Average Pooling) y la suma.

Por lo tanto, la capa de agrupación reduce el tamaño (la dimensionalidad) del mapa de características, ya que cada región de entrada produce solo un valor de salida. Esto hace que el procesamiento sea más eficiente y reduce el número de parámetros en la red.

- Capas Completamente Conectadas (Fully Connected Layers).

La Fully Connected Layer es responsable de realizar operaciones de aprendizaje y de transformación lineal en los datos de entrada, y suele estar situada al final de la red neuronal.

En una capa completamente conectada, cada neurona de la capa está conectada a todas las neuronas de la capa anterior y a todas las neuronas de la capa siguiente. Esto significa que cada entrada proveniente de la capa anterior contribuye a todas las salidas que genera la capa completamente conectada (y que pasan a la siguiente capa por tanto).

En la Fully Connected Layer se realizan dos operaciones:

1. Transformación lineal. Cada neurona de la capa completamente conectada realiza una operación de transformación lineal en los datos de entrada utilizando un conjunto de pesos (weights) y un sesgo (bias). Estos pesos y sesgos son aprendidos durante el entrenamiento de la red neuronal.

2. Función de activación. Después de la operación de transformación lineal, se aplica una función de activación no lineal a la salida de cada neurona. Esta función de activación introduce no linealidades en la red y permite a la red aprender relaciones más complejas en los datos.

Así, la salida de la capa completamente conectada es una combinación lineal de las entradas ponderadas por los pesos y sesgos aprendidos, seguida de la aplicación de la función de activación.

Las capas completamente conectadas se utilizan comúnmente en tareas de clasificación y regresión en las que los datos de entrada pueden ser representados como vectores de características. Por ejemplo, en el caso de la clasificación de imágenes, las características extraídas por las capas convolucionales pueden ser aplanadas y pasadas a través de una capa completamente conectada para realizar la clasificación final.

Las CNN que trabajan con imágenes utilizan capas convolucionales 2D, pero si queremos procesar datos unidimensionales (series temporales, señales de audio, texto...) una opción muy interesante es la denominada Red Neuronal Convolucional 1D (CNN 1D).

En las convoluciones unidimensionales, en lugar de convoluciones en dos dimensiones sobre una matriz de píxeles, las convoluciones 1D se realizan a lo largo del vector unidimensional de entrada.

Las CNN 1D son especialmente efectivas en el análisis de series temporales, donde pueden capturar patrones temporales complejos en los datos.

Y, al igual que en otras arquitecturas de CNN, las CNN 1D se benefician del Transfer Learning.

3.3. Recurrent Neural Network (RNN)

Red Neuronal Recurrente.

Hemos visto que las CNN están muy orientadas a datos de entrada matriciales. Son excelentes para procesado de imágenes.

Las RNN, en cambio, son un tipo de arquitectura de red neuronal especialmente diseñada para modelar datos secuenciales (texto, series temporales y similares).

Tampoco son expresamente modelos generativos.

En los datos secuenciales, cada dato depende del anterior o de los anteriores.

En las RNN, la información no fluye sólo en modo feedforward, hacia delante.

Las RNN tienen conexiones retro-alimentadas que les permiten mantener y utilizar información de estados anteriores en la secuencia. Esto les da a las RNN la capacidad de modelar dependencias temporales en los datos.

Es decir, las RNN tienen celdas de MEMORIA (celdas recurrentes).

La celda de memoria toma la entrada actual y el estado anterior como entrada y produce una salida y un nuevo estado interno.

Las RNN procesan datos secuenciales uno a uno, en cada paso de tiempo. En cada paso, la entrada actual se alimenta a la red junto con el estado anterior, y la red actualiza su estado interno y produce una salida. Esta salida se puede utilizar como entrada para el siguiente paso de tiempo.

El problema es que las RNN tienen memoria a corto plazo. Y en ocasiones necesitamos memoria a más largo plazo para captar dependencias alejadas en el tiempo.

3.4. (Long Short-Term Memory (LSTM)

Es una variante de las RNN que tiene memoria a largo plazo. No son modelos generativos.

Para ello tiene una serie de modificaciones sobre la estructura de una RNN convencional:

1. Unidades de Memoria. En lugar de usar una simple unidad de memoria como en las RNN estándar, las LSTM utilizan una estructura más compleja de celdas de memoria (se les llama expresamente unidades LSTM).

Cada unidad LSTM tiene una estructura interna que consta de una celda de memoria principal y tres puertas (gate): la Puerta de Entrada (Input Gate), la Puerta de Olvido (Forget Gate) y la Puerta de Salida (Output Gate).

- Puerta de Entrada (Input Gate). Esta puerta decide qué nueva información debe ser almacenada en la celda de memoria. Toma como entrada la propia entrada actual y el estado anterior, y produce un vector de activación que se multiplica elemento por elemento con una candidata de celda de memoria generada a partir de la entrada actual. La información resultante se suma a la celda de memoria actualizada.

- Puerta de Olvido (Forget Gate). Esta puerta decide qué información debe ser olvidada o mantenida en la celda de memoria. Toma como entrada la propia entrada actual y el estado anterior, y produce un vector de olvido que se multiplica elemento por elemento con el contenido actual de la celda de memoria para determinar qué información debe ser olvidada.

- Puerta de Salida (Output Gate). Esta puerta determina qué parte de la información almacenada en la celda de memoria debe ser la salida de la unidad LSTM. Toma como entrada la propia entrada actual y el estado anterior, y produce un vector de salida que se multiplica elemento por elemento con la versión filtrada del contenido actual de la celda de memoria.

Esta estructura de puertas y celdas de memoria, es la que permite a las LSTM aprender y recordar dependencias a largo plazo en las secuencias de datos.

3.5. Gated Recurrent Unit (GRU)

Las GRU son otra variante de las RNN con memoria a largo plazo. No son modelos generativos.

Se desarrollaron para abordar algunos de los problemas de las LSTM. Sobre todo la complejidad computacional que tienen las LSTM.

La unidad de memoria en la GRU es más sencilla que en el caso de las LSTM. En una GRU, la Puerta de Olvido (Forget Gate) y la Puerta de Entrada (Input Gate) están combinadas en una sola puerta, denominada Puerta de Actualización (Update Gate).

Esta puerta determina cuánta información de la celda de memoria anterior debe mantenerse y cuánta información de la nueva entrada debe integrarse. Toma como entrada la entrada actual y el estado anterior, y produce un vector de actualización que controla la proporción de información pasada y futura que se mantiene.

Algunas variantes de GRU incluyen una Puerta de Reinicio (Reset Gate) adicional que controla cuánta información de la celda de memoria anterior se debe olvidar. Esta puerta permite a la GRU adaptarse mejor a diferentes tipos de secuencias y mantener la capacidad de aprender dependencias a largo plazo.

Usando el vector de actualización calculado por la puerta de actualización, el estado de la celda de memoria se actualiza mediante una interpolación lineal entre el estado anterior y la candidata de celda de memoria generada a partir de la entrada actual.

La gran ventaja de las GRU frente a las LSTM es que su estructura más simple puede hacer que sean más fáciles de entrenar y más eficientes computacionalmente en algunas aplicaciones.

3.6. Generative Advesarial Network (GAN)

Redes Generativas Adversariales (GAN). Esta arquitectura es expresamente un modelo generativo. Es decir, sólo se utiliza para generar datos nuevos (y realistas).

Nos parece una arquitectura muy original y muy divertida.

Hoy (marzo de 2024) tal vez se considere antigua, pero lo cierto es que fue uno de los primeros grandes resultados dentro de los modelos generativos.

Esta arquitectura consta de dos redes neuronales que compiten entre sí:

- Un **discriminador**, que dictamina si una imagen es 'Real' o 'Fake'. Tiene que hacer bien su trabajo, claro está.

- Un **generador**. Su función es engañar al discriminador. Este generador produce imágenes (evidentemente son todas 'Fake' porque las ha generado él). Tiene que generar imágenes tan reales que el discriminador las cataloga como 'Real' cuando en realidad son 'Fake'.

Una vez conseguido esto, ya tenemos un generador que es capaz de generar imágenes realistas, con la ayuda inestimable de ese discriminador.

Por lo tanto, estas redes tienen un proceso de entrenamiento adversarial, de ahí su nombre.

Durante el entrenamiento, el generador y el discriminador se entrenan simultáneamente en un juego adversarial donde compiten uno contra el otro.

El generador intenta generar muestras que sean indistinguibles de las muestras reales para engañar al discriminador, mientras que el discriminador intenta mejorar su capacidad para distinguir entre muestras reales y generadas.

El parámetro fundamental en el entrenamiento de una GAN es la función de pérdida adversarial.

Esta función consiste en dos componentes:

- La pérdida del generador. La pérdida del generador se calcula para minimizar la probabilidad de que el discriminador clasifique incorrectamente las muestras generadas como falsas. Es decir, el generador funciona bien cuando el generador clasifique todas sus muestras como verdaderas.

- La pérdida del discriminador. Al contrario que la anterior, la pérdida del discriminador se calcula para maximizar la probabilidad de clasificar correctamente las muestras generadas y reales. Es decir, el discriminador funciona bien cuando reconoce todos los datos verdaderos como tales, y como falsos todos las que ha creado el generador.

A medida que el generador y el discriminador se entrenan de forma adversarial, el generador aprende a generar datos cada vez más realistas que capturan las características estadísticas y estructurales del conjunto de datos de entrenamiento original. Esto permite a las GAN generar imágenes, música, texto y otros tipos de datos que son indistinguibles de los datos reales.

Las GAN se utilizan en una amplia variedad de aplicaciones, incluyendo la generación de imágenes realistas, la mejora y el aumento de datos, la traducción de estilo de imagen, la generación de música y arte, la síntesis de voz, la creación de caras sintéticas y mucho más.

En cualquier caso, su aplicación más famosa es la generación de imágenes realistas.

3.7. Autoencoder

Los Autoencoders son un tipo especial de red neuronal que se utilizan para aprender representaciones eficientes de datos de entrada, lo que los hace útiles en tareas de reducción de dimensionalidad, denoising (eliminación de ruido), y generación de datos.

Por lo tanto, pueden ser modelos generativos también.

Veamos su estructura básica. Un Autoencoder consta de dos partes principales:

- Codificador (encoder). El codificador toma como entrada los datos de entrada y los transforma en una representación comprimida, llamada espacio latente o embedding.

- Decodificador (decoder). El decodificador luego toma esta representación comprimida y la reconstruye de nuevo a la forma original de los datos de entrada.

Es decir, el Autoencoder primero comprime los datos de entrada y luego los reconstruye intentando volver a su forma original.

Durante el entrenamiento, el Autoencoder intenta minimizar la diferencia entre los datos de entrada y los datos de salida reconstruidos. Esto se hace ajustando los pesos de las capas del codificador y del decodificador utilizando técnicas de optimización como el descenso de gradiente.

Una de las aplicaciones más comunes de los Autoencoders es la reducción de dimensionalidad. Al restringir el tamaño de la capa oculta (espacio latente), el Autoencoder aprende a representar los datos de entrada de manera más compacta, lo que puede ser útil para visualización de datos, compresión de datos y eliminación de ruido.

Además de la reducción de dimensionalidad, los Autoencoders también pueden utilizarse para generar datos nuevos y realistas. Esto se hace alimentando vectores de ruido aleatorio en el decodificador y generando datos sintéticos a partir de la representación aprendida en el espacio latente.

Vamos a ver cómo funciona un Autoencoder:

1. Preparación del Autoencoder. Supongamos un Autoencoder entrenado previamente con muchas imágenes de caras humanas. Durante el entrenamiento, este Autoencoder ha aprendido a comprender las características importantes de las caras humanas y a representarlas de manera eficiente en un espacio de menor dimensión llamado espacio latente.

2. Generación de nuevas caras. Ahora, cuando queramos generar una nueva cara que se parezca a las caras humanas, en lugar de darle una cara real al Autoencoder, le damos un conjunto de números aleatorios (ruido) como entrada. Este conjunto de números aleatorios no tiene ninguna relación directa con ninguna cara en particular. Es sólo ruido.

3. Proceso de generación. El Autoencoder toma estos números aleatorios y los procesa a través de su red neuronal. Utiliza lo que ha aprendido durante el entrenamiento para intentar reconstruir una cara humana basada en esos números aleatorios. Como ha sido entrenado con muchas caras humanas, el aspecto que tiene una cara humana en general.

4. Resultado. La salida del Autoencoder es una nueva cara que ha sido generada a partir del ruido aleatorio. Esta cara será similar a las caras humanas reales, pero no será idéntica a ninguna de ellas. Sin embargo, es realista y podría parecerse a una cara humana genuina.

En el proceso de generación de datos utilizando un Autoencoder, se le proporciona al decodificador del Autoencoder un conjunto de números aleatorios, que pueden considerarse ruido. Estos números aleatorios no tienen ninguna relación directa con los datos originales utilizados durante el entrenamiento del Autoencoder.

El propósito de alimentar el decodificador con estos números aleatorios es permitir que el Autoencoder genere nuevos datos que sean diferentes de los datos de

entrenamiento, pero que sigan siendo realistas en el contexto de los datos originales. A través de su entrenamiento previo, el Autoencoder ha aprendido a capturar las características y estructuras importantes de los datos de entrenamiento en su espacio latente, y puede utilizar esta información para generar nuevas muestras que se asemejen a los datos originales.

Por lo tanto, al alimentar el Autoencoder con ruido aleatorio, se le está pidiendo que genere datos nuevos y 'creativos', basándose en su comprensión de las características del conjunto de datos original. Esto es lo que hace que los Autoencoders sean tan útiles para la generación de datos sintéticos en el aprendizaje profundo.

En resumen, el Autoencoder puede generar nuevas caras humanas realistas a partir de números aleatorios simplemente porque ha sido entrenado para entender cómo se ven las caras humanas y puede recrearlas utilizando su conocimiento previo.

Existen varias variaciones de Autoencoders, incluyendo Autoencoders Convolucionales (CAE) para datos de imágenes y Autoencoders Recurrentes (RAE) para datos de secuencias temporales. Variational Autoencoders (VAE), que pueden aprender representaciones más suaves y continuas y generar datos nuevos y realistas

3.8. Transformer

Esta arquitectura se desarrolló inicialmente para tareas de procesamiento de lenguaje natural (NLP), pero ha demostrado ser altamente efectiva en una variedad de aplicaciones secuenciales, como la traducción automática, la generación de texto, la síntesis de voz y el reconocimiento de voz. Incluso en visión por computadora o en manejo de datos secuenciales, temporales.

Todo ello gracias a su alta capacidad para capturar dependencias a largo plazo en los datos de entrada.

Actualmente (marzo 2024) es la arquitectura por excelencia sobre todo en cuanto a generación de textos. Pero también pueden ser utilizados para tareas de discriminación, como la clasificación de texto o la extracción de información.

La gran novedad de esta arquitectura es que introduce el mecanismo de ATENCIÓN[2].

Veamos que qué consiste este mecanismo de atención.

Supongamos que estamos leyendo un texto y tratando de entenderlo. Cuando encontramos una palabra, nuestra mente busca otras palabras relacionadas para entender mejor el contexto. Por ejemplo, si vemos la palabra 'perro', automáticamente pensaremos en palabras relacionadas con los perros como 'peludo', 'ladra', 'mascota', etc.

En los Transformers, este proceso se llama 'atención'.

Este mecanismo pretende funcionar de manera similar a cómo nuestra mente piensa palabras relacionadas cuando leemos. Pero, en lugar de palabras, los Transformers procesan secuencias de palabras o tokens.

Veamos algunos conceptos interesantes para comprender el funcionamiento de los Transformers, a grandes rasgos.

Auto-Atención (Self-Attention). La característica principal de los Transformers es el mecanismo de Self-Attention. En lugar de depender de conexiones secuenciales como en las RNN, el Transformer procesa todas las palabras (o tokens) de la entrada simultáneamente, en paralelo. La Self-Attention permite a cada palabra en la secuencia 'atender' a todas las demás palabras, calculando una puntuación de atención que indica la importancia relativa de cada palabra para la palabra actual.

La puntuación de atención en la Self-Attention se calcula utilizando una función de atención, que generalmente incluye las tres componentes consulta, clave y valor.

Estos componentes se multiplican para calcular los pesos de atención, que luego se aplican a los valores para producir la salida ponderada final.

Self-Attention Layers. Cada capa de Auto-Atención en un Transformer se compone de múltiples cabezas de atención (Multi-Head Attention), cada una de las cuales calcula una puntuación de atención diferente. Estas puntuaciones de atención se utilizan para ponderar las representaciones de las palabras en la secuencia, permitiendo que el modelo se enfoque en diferentes aspectos de la entrada en diferentes contextos.

Codificador y decodificador. Los Transformers constan de dos partes principales: un codificador y un decodificador. El codificador procesa la secuencia de

entrada y extrae características relevantes, mientras que el decodificador genera la salida basada en estas características. Ambos están compuestos por múltiples capas de bloques de Self-Attention y capas de alimentación directa (Feedforward).

Conexiones residuales y normalización por capas. Para facilitar el entrenamiento, los Transformers utilizan conexiones residuales en cada capa individual. Esto es, la salida de cada capa individual se suma a la entrada original antes de pasar a la siguiente capa.

La normalización es una manera de estandarizar la distribución de los valores de activación en cada capa. Esta estandarización puede ayudar a mejorar la convergencia del entrenamiento y la capacidad de generalización del modelo.

Transfer Learning y pre-entrenamiento. Los Transformers se pueden preentrenar en grandes conjuntos de datos no supervisados utilizando tareas como el auto-rellenado de máscaras (BERT) o la predicción de la siguiente oración (GPT), y luego ajustarse o afinarse en tareas específicas con conjuntos de datos más pequeños y etiquetados.

El proceso de atención en los Transformers sigue estos pasos:

1. Consulta, Claves y Valores. Supongamos una frase. Para cada palabra (o token) en esa oración, el modelo Transformer genera tres componentes: una 'consulta', unas 'claves' y unos 'valores'. La consulta es la palabra que estamos tratando de entender, las claves son las palabras que están relacionadas con esa palabra, y los valores son la información real que está asociada con esas palabras.

2. Puntuación de Atención. Después, el modelo compara la consulta con todas las claves para cuantificar el grado de cercanía entre ellas, si están estrechamente relacionadas o no tanto. Esto se llama 'puntuación de atención'. Cuanto más relacionadas estén la consulta y una clave, mayor será la puntuación de atención.

3. Softmax y Pesos de Atención. Posteriormente, estas puntuaciones de atención pasan por una función matemática llamada Softmax, que básicamente convierte las puntuaciones de atención en números entre 0 y 1, asegurándose de que sumen 1 en total. Estos números se llaman 'pesos de atención' y muestran cuánta importancia tiene cada palabra en relación con la consulta.

4. Atención Ponderada. Finalmente, se multiplican los valores por los pesos de atención para obtener una especie de 'promedio ponderado' de todos los valores. Esto nos da una representación contextualizada de la palabra original, lo que significa que tenemos una mejor comprensión de esa palabra en su contexto.

A modo de resumen, los Transformers utilizan codificadores posicionales para etiquetar elementos de datos. Las unidades de atención siguen estas etiquetas, calculando una especie de mapa algebraico de cómo cada elemento se relaciona con los demás.

Por lo tanto, podemos decir que el Transformer aprende contexto, o lo intenta.

Tenemos dos orejas y una sola boca, justamente para oír más y hablar menos

— Zenón de Citio, *La más antigua de sus obras es con mucha probabilidad La República. Compuesta cuando aún estudiaba con Crates. Fue escrita en respuesta a La República de Platón y otorga propuestas totalmente opuestas a esta.*

4

Requisitos para Preparar un Buen Dataset

Índice

4.1. ¿Cómo Debe Ser un Buen Dataset?

Lo primero y más importante: un modelo Deep Learning fiable y robusto debe entrenar CON MILES DE DATOS.

Si no entrena con miles de datos, no va a ser un modelo fiable. Puede que aparente ser robusto, pero no será fiable.

Además de disponer de miles de datos, estos deben ser equilibrados ente todas las categorías. Tenemos 20.000 imágenes de perros y gatos, esto está bien. Pero, de esas 20.000, 19.800 son imágenes de perros y sólo 200 imágenes son de gatos. El modelo no va a funcionar bien en el caso de la categoría de gatos.

Y además, las imágenes deben incluir todas las casuísticas de manera suficiente. Este punto es extremadamente importante.

Supongamos un clasificador de motos de nieve. Detecta de una manera espectacular motos de nieve. No importa la distancia o la escasa nitidez de la imagen, siempre detecta una moto de nieve.

Hasta que un día descubrimos que sólo detecta bien motos de nieve si el suelo está nevado. Este tipo de sucesos es muy habitual al trabajar con redes neuronales, tengan siempre cuidado.

Una moto de nieve en un remolque en verano, no es una moto de nieve para esa red neuronal.

¿Cómo es posible? Porque probablemente en su entrenamiento, todas las imágenes de motos de nieve incorporaron un suelo nevado. Y la red neuronal estableció una correlación directa entre ambos hechos. Para esta red, su conclusión fue algo así como que sólo puede ser moto de nieve si el suelo es blanco.

En su entrenamiento faltaron imágenes de la moto de nieve sobre un remolque, motos de nieve en verano, etc.

Esta es la importancia de contemplar todas las casuísticas en un entrenamiento.

Para incluir adecuadamente todas las casuísticas, tenemos que conocer con precisión la representatividad de los datos utilizados. Los datos deben capturar de manera precisa y completa la variabilidad y diversidad presentes en el fenómeno en cuestión.

Supongamos ahora un detector de cisnes. Supongamos que el porcentaje de cisnes blancos es un 99,5 % y el de cisnes negros es de un 0,5 %. Aparentemente, la cantidad de cisnes negros no es representativa y podemos tender a obviarla. Pero si queremos un buen clasificador de cisnes, queremos que detecte los cisnes de todas las especies. En Australia y Nueva Zelanda hay una especie denominada *Cygnus atratus*. Los cisnes que pertenecen a esta especie son negros. Por lo tanto, si no consideramos los cisnes negros, estamos dejando fuera a una especie entera de cisne, la principal de uno de los cinco continentes de la Tierra. De repente este 0,5 % tiene más importancia de la que aparentaba, ¿verdad? Cuidado con dejarnos llevar sólo por los números.

Por lo tanto, tenemos que conocer MUY BIEN la representatividad de los datos utilizados. Recomendamos dedicar tiempo a conocer el ámbito de los datos. Si trabajamos con imágenes de cisnes, necesitamos conocer qué especies de cisnes

hay, cuáles son sus peculiaridades morfológicas, etc. Y después, empezaremos a prospectar datos.

Puede ser que haya características o peculiaridades que a priori parezcan superfluas. Recomendamos en este fase de investigación y análisis considerar todas. Sólo cuando tengamos una imagen fiel del ámbito de trabajo, un conocimiento profundo del mismo, seremos capaces de discriminar adecuadamente.

Tenemos que tener datos de todas las categorías disponibles, existentes. Recomendamos incluir las más nimias o improbables.

Siempre evitaremos la tendencia a trabajar únicamente con ciertas categorías direccionadas. Si sólo tenemos 100 imágenes de *Cygnus atratus*, en lugar de eliminar esta especie del dataset, recomendamos conseguir como sea más imágenes. Sólo así tendremos un modelo realmente eficiente detectando cisnes.

Otro asunto, tendremos datos suficientes de todas las casuísticas. Miren estas fotos:

(a) Esto es un pájaro **(b)** Esto también es un pájaro

Figura 4.1: Ejemplo de casuísticas. Fotografías realizadas por Ana Guerrero Tamayo.

Queda claro lo que queremos decir, ¿verdad? Por favor, no olviden nunca esto.

Merece la pena dedicar tiempo a recopilar el mejor y más completo dataset frente a preparar un modelo lo antes posible.

4.2. Cuidado en el Uso de Transfer Learning

Es una técnica de aprendizaje automático donde un modelo pre-entrenado en una gran cantidad de datos y para una tarea específica es adaptado o transfiere su conocimiento para una tarea similar o relacionada.

El Transfer Learning, o aprendizaje por transferencia, es una técnica en el aprendizaje automático y la inteligencia artificial donde un modelo preentrenado en una tarea se utiliza como punto de partida para resolver una tarea relacionada pero diferente. Aquí tienes un desglose detallado de cómo funciona el Transfer Learning:

1. Pre-entrenamiento del modelo base. En el Transfer Learning, primero se entrena un modelo en una tarea relevante, normalmente utilizando un conjunto de datos extraordinariamente grande, muy abundante y general. Es decir, con millones de datos de miles de categorías. Este modelo pre-entrenado se denomina 'modelo base' o 'modelo pre-entrenado'. En la práctica, estos modelos pre-entrenados están disponibles y simplemente los utilizamos para un nuevo propósito ya que no tenemos medios para fabricarnos nuestro propio modelo pre-entrenado normalmente.

2. Reutilización del conocimiento. Una vez que el modelo base ha sido pre-entrenado en una tarea, se puede utilizar para inicializar los pesos de un nuevo modelo que se entrenará en una tarea específica. Los pesos del modelo pre-entrenado se utilizan como punto de partida, y luego se ajustan o 'afinan' durante el entrenamiento en la nueva tarea.

3. Tareas de destino relacionadas. El Transfer Learning es especialmente efectivo cuando la tarea de destino está relacionada con la tarea original para la que se pre-entrenó el modelo base. Por ejemplo, un modelo pre-entrenado en reconocimiento de imágenes puede ser ajustado para tareas específicas como detección de objetos, clasificación de enfermedades médicas o reconocimiento de gestos.

4. Adaptación de la arquitectura del modelo. Dependiendo de la similitud entre la tarea original y la tarea de destino, es posible que sea necesario adaptar la arquitectura del modelo pre-entrenado. Esto puede implicar agregar o eliminar capas, modificar la estructura de las capas existentes o cambiar los parámetros de entrenamiento, como la tasa de aprendizaje.

5. Ajuste fino del modelo (fine-tuning). Una vez que se ha inicializado el nuevo modelo con los pesos del modelo preentrenado, se entrena el modelo fino en la tarea de destino utilizando un conjunto de datos más pequeño y específico. Durante este proceso de ajuste fino, los pesos del modelo se actualizan para adaptarse mejor a la nueva tarea.

El Transfer Learning tiene varios beneficios, incluyendo la capacidad de aprovechar el conocimiento previo aprendido en tareas relacionadas, la capacidad de entrenar modelos efectivos con conjuntos de datos más pequeños y la reducción del tiempo y los recursos necesarios para entrenar modelos desde cero.

En resumen, el Transfer Learning permite reutilizar el conocimiento aprendido en una tarea para resolver nuevas tareas. Esto permite acelerar el proceso de entrenamiento y mejorar el rendimiento del modelo en tareas con conjuntos de datos limitados.

Es una herramienta valiosa, pero no debemos depender exclusivamente de ella sin realizar una validación y verificación adecuadas.

Es importante realizar una validación y verificación adecuadas para garantizar que el modelo transferido se ajuste y funcione correctamente para la nueva tarea específica. Esto implica ajustar hiperparámetros, seleccionar capas adecuadas, evitar el Overfitting y el Underfitting, y evaluar el rendimiento del modelo de manera integral.

Si tenemos un dataset con 100 imágenes de perros y 100 imágenes de gatos, por mucho Transfer Learning que apliquemos, NUNCA TENDREMOS UN MODELO FIABLE.

El Transfer Learning nunca puede sustituir un dataset deficiente.

4.3. Cuidado con la Generación de Muestras Sintéticas

Podemos generar muestras sintéticas de varias maneras. En el caso de imágenes, es habitual utilizar Data Augmentation. Ante una determinada imagen, esta técnica consiste en rotarla, ampliarla, reducirla, etc. Con cada una de estas modificaciones,

o incluso aplicando varias simultáneamente, obtenemos una imagen nueva. Así podemos obtener varias nuevas imágenes modificadas de la original, que se pueden considerar nuevos datos.

Debemos tener en cuenta lo siguiente: por mucha modificación directa sobre una imagen que hagamos, el contenido crítico de la imagen es el mismo, no cambia el trasfondo.

Veamos un ejemplo de aplicación de Data Augmentation sobre una imagen:

(a) Imagen Original **(b)** Imagen con Data Augmentation

Figura 4.2: Ejemplo de Data Augmentation. Fotografías realizadas por Ana Guerrero Tamayo

Hemos aplicado Data Augmentation simplemente girando una imagen con dos gatos. Para una red neuronal, son dos imagenes diferentes. Pero el contenido de esas dos imágenes es el mismo. Son los mismos dos gatos.

Si aplicamos Data Augmentation sobre una imagen microscópica de un determinado virus, podemos obtener decenas de imágenes. Pero la foto original del virus es la misma.

También podemos generar muestras sintéticas utilizando modelos generativos. Por ejemplo, podemos utilizar una red GAN o un Autoencoder para producir imágenes totalmente nuevas. O generar nuevos textos utilizando modelos de generación de lenguaje natural.

En estos casos el riesgo puede ser aún mayor. Si el modelo no funciona del todo bien, todas las muestras generadas por el modelo pueden incluir pequeños defectos

que las hagan escasamente operativas. En el caso de textos, por ejemplo, todos los textos generados por un mismo GPT (Generative Pre-trained Transformer) pueden incluir determinadas expresiones, vocabulario o giros gramaticales propios y característicos de ese modelo. Y empobreceremos la variedad del lenguaje.

Todas estas salvedades pueden ser especialmente graves si trabajamos en un ámbito mayormente desconocido. Por ejemplo: generación sintéticas de genomas. Puede ser un divertimento pero en ningún caso puede utilizarse para una aplicación seria en el ámbito de la genómica, dado el margen de error no asumible por desconocido. Aún apenas conocemos el significado del genoma. ¿Cómo vamos a reproducirlo si no sabemos qué contiene ni cómo funciona?

Podemos pensar que, limitando la aplicación de estas técnicas a un aumento de datos del, por ejemplo, 20 %, podemos subsanar el riesgo de generar excesivas muestras sintéticas.

En nuestra opinión, si vamos a aplicar modelos Deep Learning en ámbitos serios (no es lo mismo clasificar tumores que clasificar perros y gatos), de alta importancia y/o alto desconocimiento, no debemos aplicarlas en ningún caso. Podemos introducir sesgos en el modelo.

Deberemos dedicar todos nuestros esfuerzos a obtener más y mejores muestras reales, de acuerdo a lo expresado en la sección como debe ser un buen dataset.

Y si no es posible obtener un número suficiente de muestras para entrenar el modelo en condiciones, simplemente no deberemos aplicarlo. Esto lo repetiremos muchas veces a lo largo de este libro: NO TODO ES IA.

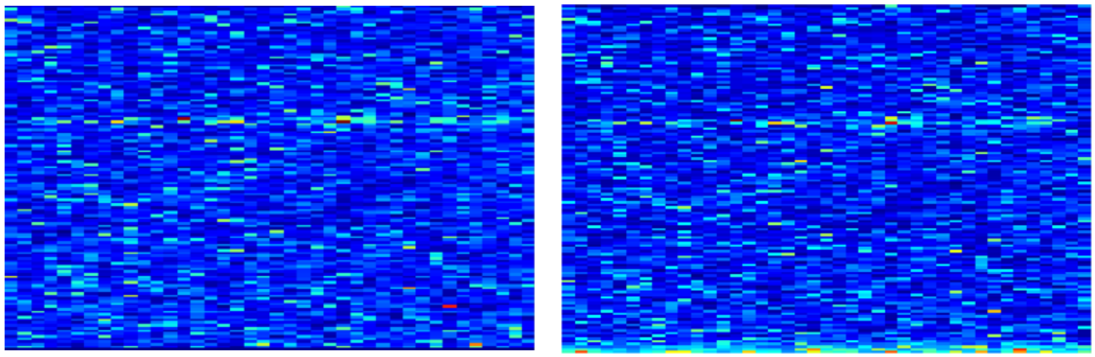
Debemos cuestionar la validez de todos los modelos entrenados con un dataset que incluya muestras artificiales. Insistimos, en aplicaciones en ciencias puras y ciencias aplicadas. En 'aplicaciones serias'.

4.4. Las Redes Neuronales a Veces Son un Misterio

A veces las redes neuronales nos pueden parecer inestables. Esto no significa que realmente lo sean, simplemente nos lo parecen porque desconocemos su

funcionamiento intrínseco. Sólo somos capaces de ver el resultado final pero no sabemos cómo han llegado hasta ahí.

Veamos un ejemplo real.



(a) La red Funciona Bien **(b)** La Red No Ha Aprendido Nada

Figura 4.3: Dos Datos Aparentemente Iguales, Dos Comportamientos Diferentes. Imágenes generadas por Ana Guerrero Tamayo.

Tenemos dos datasets formados por espectrogramas de señales digitales. Entrenando un mismo modelo con el dataset representado por la imagen de la izquierda, obtuvimos resultados positivos y robustos.

Sin embargo, con el dataset representado por la imagen de la derecha, ese mismo modelo, con un mismo entrenamiento exactamente replicado, no aprendió nada. Las clasificaciones de ese modelo eran absolutamente aleatorias. ¿Por qué? A día de hoy seguimos sin saberlo realmente.

Si nos fijamos, vemos que en la parte inferior de la figura 4.3b hay una zona coloreada que no existe en la figura 4.3a. Muy probablemente ese sea el motivo de este comportamiento anómalo del modelo con el dataset de imágenes con el formato de la derecha.

A través de una metodología basada totalmente en ensayo y error, detectamos que la razón se encontraba en las componentes de baja frecuencia (línea inferior de ambas imágenes). Filtrando adecuadamente los espectrogramas para que desaparezca esa carga en las bajas frecuencias, desapareció el problema.

No podemos saber las implicaciones que tiene esa pérdida de información sobre el conocimiento real. Al realizar un filtrado, perdemos inevitablemente información.

Hemos logrado un modelo que funcione, podremos extraer conclusiones, pero todo eso será siempre sin poder incluir la información filtrada.

Si trabajamos en un ámbito desconocido, estamos perdiendo datos que ni siquiera sabemos si son realmente relevantes o no. Estamos introduciendo un sesgo por omisión de información.

El único consejo que podemos dar en este asunto es que es importante medir bien el riesgo en la medida de lo posible. Analizar muy bien qué estamos perdiendo en el camino y las consecuencias de no poder trabajar con ello.

La ciencia se compone de errores, que a su vez, son los pasos hacia la verdad

— Julio Verne, *aunque es famoso por sus predicciones científicas visionarias en sus obras de ciencia ficción, en realidad tenía aversión hacia la tecnología emergente de su época.*

5 El Método Científico

Índice

5.1. ¿Qué es el Método Científico?

No pretendemos analizar y definir exhaustivamente el método científico. Simplemente queremos dar unas nociones para que todos aquellos ingenieros en computación que desarrollen aplicaciones Machine Learning y Deep Learning en el ámbito de la ciencia, sepan de qué estamos hablando.

El método científico es un proceso sistemático, riguroso y exhaustivo orientado a extraer conocimiento veraz.

Para ello consta de una serie de etapas principales:

1. Observación. La observación está fuertemente influenciada por la **subjetividad**. El observador es un ser humano, con sus propios conocimientos, experiencias, prejuicios, opiniones, etc. Por lo tanto, su adquisición y registro de información no son imparciales. Precisamente esto es lo que el método científico busca corregir.

2. Hipótesis. Una declaración no verificada (con una conclusión o predicción) que puede ser confirmada o refutada.

3. Experimentación. Se busca reproducir las condiciones del fenómeno o caso de estudio en un ambiente controlado (generalmente un laboratorio) y con una cantidad limitada de variables. Se debe introducir un grupo de control. A este grupo no se le elimina ni se le introduce ninguna variable en relación al fenómeno original.

4. Análisis.

5. Conclusión. Para que una conclusión sea considerada válida, todo el experimento que la produjo debe ser **replicable**.

Un apunte. Probablemente llevamos unos 2.500 años intentando conseguir una imparcialidad en la observación. Aristóteles (384-322 a.C.) y otros grandes pensadores antiguos ya intentaron minimizar el impacto de la subjetividad a través del pensamiento lógico.

Estamos en 2024 y aún no lo hemos conseguido de una forma efectiva. Por algo será...

A la vista de estas simples definiciones de las etapas del método científico, ya nos encontramos con dos salvedades para la aplicación de modelos Machine Learning y Deep Learning en el ámbito científico:

1. El principal es el relacionado con la subjetividad. Lo veremos en detalle en el Capítulo 6.

2. El segundo es el de la replicabilidad. Las redes neuronales, por la naturaleza de sus arquitecturas, nunca obtienen exactamente el mismo resultado. Los grados de acierto se moverán en un rango estrecho pero nunca serán el mismo. En condiciones

muy exigentes donde se requiere una replicabilidad absoluta, simplemente no podemos utilizar estos algoritmos.

5.2. Falacia

Es un término muy utilizado en la aplicación del método científico. Importantísimo.

Una falacia es simplemente un argumento que parece lógico, pero no lo es.

Puede originarse en limitaciones técnicas, errores metodológicos, prejuicios y sesgos cognitivos.

Veamos como ejemplo la famosa falacia del francotirador. Un francotirador dispara varios tiros aleatorios en un granero y luego dibuja un objetivo alrededor de cada uno de sus tiros. La conclusión, al ver el granero, es que es un excelente tirador, pero la premisa es falsa. Esta falacia lógica sirve para ilustrar que es posible relacionar varios elementos para llegar a la conclusión deseada, sin que esta sea necesariamente verdadera.

Las falacias son errores de razonamiento que pueden parecer convincentes, pero que en realidad son defectuosos o engañosos en su lógica. Se presentan en diferentes formas y se clasifican en varias categorías según su estructura o naturaleza.

Veamos algunos tipos de falacias y ejemplos para facilitar la comprensión de los mismos.

5.2.1. Falacias de Relevancia

Hay varios subtipos. Y todos pueden ser horribles. Exponemos una serie de ejemplos espantosos... Y reales. Por favor, no olvidemos nunca el efecto de las falacias. Veamos.

Argumentum ad populum.

Argumentar que algo es verdadero porque muchas personas lo creen o lo aceptan.

Persona A: 'La mayoría de las personas blancas en Sudáfrica apoyan el apartheid y creen que los negros son inferiores. Por lo tanto, debe ser la forma correcta de gobierno.'

Persona B: 'Pero el apartheid es una política de discriminación racial injusta y deshumanizante. No importa cuántas personas lo apoyen, sigue siendo moralmente incorrecto.'

Persona A: 'Pero si tanta gente blanca lo apoya, debe ser lo correcto. Los negros no pueden gobernarse a sí mismos adecuadamente'.

Argumentum ad verecundiam.

Apelar a la autoridad de una persona o entidad como justificación para la verdad de una afirmación.

Persona A: 'El régimen nazi ha promulgado leyes que discriminan y persiguen a los judíos. Sin embargo, muchos científicos y académicos alemanes respaldan estas leyes, por lo que deben tener fundamentos sólidos y ser justificadas.'

Persona B: 'Pero las leyes antisemitas son moralmente reprensibles y violan los derechos humanos básicos.'

Persona A: 'Los líderes nazis han consultado a expertos y académicos, y si están de acuerdo con estas políticas, debe haber razones válidas para su implementación. Además, desafiar estas políticas sería desobedecer a las autoridades legítimas del país.'

Argumentum ad hominem.

Atacar al carácter o la credibilidad de una persona en lugar de abordar el argumento que está presentando.

Persona A: 'La activista feminista X está abogando por la igualdad de género, pero es simplemente una mujer enojada que odia a los hombres. No podemos tomar en serio sus argumentos.'

Persona B: 'Pero sus argumentos están respaldados por datos y evidencia sólida sobre la desigualdad de género en nuestra sociedad.'

Persona A: 'No importa, ella claramente tiene una agenda personal y su opinión está sesgada debido a su odio hacia los hombres. No necesitamos escuchar lo que tiene que decir.'

5.2.2. Falacias de Ambigüedad

También tenemos varios subtipos. Veamos.

Falacia de la elusión.

Evitar abordar directamente un punto o una pregunta, desviando la atención hacia otro tema no relacionado.

Persona A: 'Debemos discutir cómo mejorar la educación pública en nuestro país para garantizar que todos los niños tengan acceso a una educación de calidad.'

Persona B: 'Sí, la educación es importante, pero ¿qué hay de la economía? Necesitamos enfocarnos en mejorar la economía para crear más empleos y oportunidades.'

Falacia de la equívoca.

Utilizar términos ambiguos o vagos que pueden interpretarse de múltiples maneras.

Persona A: 'Creo que deberíamos invertir en más educación para mejorar la sociedad en general. La educación es fundamental para el progreso y el desarrollo de un país.'

Persona B: 'Pero ya tenemos muchas escuelas. ¿Por qué necesitamos más educación?'

En este ejemplo, Persona B comete la falacia de la equívoca al interpretar de manera equivocada el término 'educación'. Mientras que Persona A estaba hablando sobre la necesidad de invertir en programas educativos y recursos adicionales para mejorar la calidad de la educación, Persona B interpreta el término como el número físico de escuelas existentes. Este malentendido surge debido a la ambigüedad en el uso del término 'educación' en la conversación.

5.2.3. Falacias de Falsa Causa

También podemos determinar varios subtipos.

Post hoc ergo propter hoc.

Asumir que, porque un evento ocurrió después de otro, el primero fue la causa del segundo.

Veamos un ejemplo. Durante la Edad Media, en Europa se produjo un brote de la Peste Negra, una devastadora pandemia que afectó a millones de personas y tuvo un impacto significativo en la sociedad de la época. Algunas personas podrían haber argumentado:

'Después de que se prohibiera la música y el baile en las calles, la propagación de la Peste Negra disminuyó. Por lo tanto, la prohibición de la música y el baile debe haber sido la razón de la disminución de la pandemia.'

Cum hoc ergo propter hoc.

Inferir una relación causal entre dos eventos simplemente porque ocurren juntos.

Un ejemplo. Durante la conquista de América por parte de los españoles en el siglo XVI, se registró un eclipse solar en una fecha particularmente significativa. Algunas personas podrían haber argumentado:

'Después del eclipse solar, los españoles lograron una importante victoria en la conquista de América. Por lo tanto, el eclipse debe haber sido un presagio de la victoria española.'

5.2.4. Falacias de Presunción

Algunos subtipos de estas falacias.

Falsa dicotomía.

Presentar una situación como si solo hubiera dos opciones posibles, cuando en realidad hay más.

Un ejemplo: durante una conversación sobre estilos de vida saludables, alguien comenta: 'O comes exclusivamente alimentos orgánicos y haces ejercicio todos los días, o no te preocupas por tu salud en absoluto. No hay un término medio.'

Pero sí que lo hay, ¿verdad? Cualquier término medio.

Circunstancial ad hominem.

Esta falacia se produce cuando se intenta desacreditar un argumento al señalar que la persona que lo presenta tiene intereses, circunstancias o motivaciones que podrían sesgar su punto de vista.

Imaginemos una discusión sobre la implementación de políticas ambientales más estrictas para reducir la contaminación. Una persona presenta argumentos sólidos a favor de estas políticas, destacando los beneficios para la salud pública y el medio ambiente. En lugar de abordar los argumentos directamente, otro participante señala que la persona que defiende las políticas es propietaria de acciones en una empresa de energía no renovable. Entonces, sugiere que la persona está promoviendo políticas ambientales estrictas solo para beneficiar sus inversiones en energía renovable y no porque esté genuinamente preocupada por el medio ambiente.

5.2.5. Falacias de Ignorancia

Veamos los subtipos.

Argumentum ad ignorantiam.

Argumentar que algo es cierto porque no hay evidencia de que sea falso, o viceversa.

Supongamos un país con un durísimo régimen dictatorial. Se llevan a cabo numerosas desapariciones forzadas y ejecuciones extrajudiciales. Sin embargo, el gobierno niega rotundamente su participación en estos crímenes y argumenta: 'No hay pruebas concluyentes que demuestren nuestra implicación en estas desapariciones y ejecuciones, por lo tanto, no somos responsables de ellas'.

En este caso, el gobierno está utilizando la falta de pruebas concretas que demuestren su participación en los crímenes como evidencia de su inocencia. Sin embargo, la ausencia de pruebas no necesariamente prueba la inocencia del gobierno, especialmente si se están ocultando pruebas o si las investigaciones están siendo obstruidas. Utilizar la falta de evidencia como evidencia de inocencia es una falacia de Argumentum ad ignorantiam, ya que se basa en la falta de evidencia para afirmar una conclusión. Esto puede ser particularmente impactante y desgarrador, ya que niega la responsabilidad por atrocidades cometidas durante períodos oscuros de la historia reciente.

Argumentum ad baculum.

Utilizar la amenaza de fuerza o violencia para respaldar un argumento.

Veamos un ejemplo desgraciadamente muy actual.

Durante el control de un régimen religioso extremista, el líder declara públicamente: 'Aquellos que desobedezcan las leyes de nuestra religión o se opongan a nuestro gobierno serán castigados severamente. La apostasía, la blasfemia y cualquier forma de desviación ideológica serán consideradas crímenes graves y serán castigadas con la máxima severidad. No dudaremos en aplicar penas como la flagelación, la amputación o incluso la pena de muerte para proteger la pureza de nuestra fe y mantener el orden en nuestra sociedad'.

En este caso, el líder religioso utiliza la amenaza de castigos brutales, basados en la interpretación extremista de su ley religiosa, como medio para imponer el control y la sumisión de la población. Este tipo de Argumentum ad baculum se emplea para instaurar el miedo y la obediencia a través de la aplicación de castigos físicos extremadamente severos, con el objetivo de mantener la autoridad del régimen y reprimir cualquier forma de oposición o disidencia.

5.2.6. Falacias de Inducción

Esta falacia ocurre cuando se saca una conclusión generalizada basada en un número limitado de casos o ejemplos específicos. Por ejemplo, si alguien concluye que todos los cisnes son blancos porque ha visto algunos cisnes blancos, eso sería una falacia de inducción. La premisa de que todos los cisnes son blancos se basa en una muestra limitada y puede haber excepciones no observadas.

La falacia de inducción, también conocida como generalización apresurada, ocurre cuando se saca una conclusión generalizada basada en un número limitado de casos o ejemplos específicos. En otras palabras, se asume que algo es verdadero para todos los casos basándose únicamente en evidencia anecdótica o en una muestra insuficiente. En definitiva, una premisa es relevante y apoya la conclusión, pero no la garantiza.

Esta falacia se produce cuando se infiere una regla general a partir de una observación limitada, sin considerar la posibilidad de que haya excepciones o de que se necesite más evidencia para respaldar la conclusión.

Por ejemplo, si alguien conoce a algunas personas de una determinada ciudad que son muy amables, podría concluir apresuradamente que todas las personas de esa ciudad son amables. Esta conclusión generalizada no tiene en cuenta la diversidad de personas que pueden existir en una ciudad y se basa únicamente en una muestra limitada.

Otro ejemplo, una persona en el polo norte, que solo ha visto osos polares en su vida, puede concluir que todos los osos son blancos sin que esto sea verdadero.

Otro ejemplo, espantoso. Durante la década de 1950 en Estados Unidos, después de que algunos individuos afroamericanos fueran arrestados por delitos menores en determinadas áreas urbanas, algunos miembros de la comunidad blanca concluyeron que todos los afroamericanos eran propensos a la delincuencia. Basándose únicamente en estos incidentes aislados, se generalizó erróneamente que todos los afroamericanos compartían las mismas tendencias criminales. Esta generalización injusta e irracional alimentó la discriminación sistémica, el racismo y la segregación racial en la sociedad estadounidense, perpetuando una percepción negativa y estereotipada de la comunidad afroamericana.

Aquí podemos meter miles de estereotipos. Todos los andaluces son graciosos, todos los vascos son etarras, todos los catalanes son tacaños, las mujeres son unas histéricas... Miles de estereotipos, muy ofensivos a veces.

Y que se trasladan inconscientemente a nuestro dataset...

Es importante tener en cuenta que la inducción puede ser útil y válida cuando se basa en una muestra representativa y suficiente de casos, pero la falacia de inducción ocurre cuando se generaliza demasiado rápido o sin suficiente evidencia. Para evitar esta falacia, es importante buscar evidencia sólida y considerar todas las posibles excepciones antes de sacar conclusiones generalizadas.

5.3. Falacias en Computación

El motivo de introducir toda una sección dedicada a filosofía en este capítulo es precisamente aprender a detectar falacias. Tenemos que detectar falacias en nuestras metodologías de trabajo para poder conseguir aplicar algoritmos Machine

Learning y Deep Learning con la mayor rigurosidad posible. Si trabajamos en un entorno científico (puro o ciencia aplicada), tenemos que ser capaces de aplicar el método científico de la manera más exhaustiva y veraz posible. Y por eso tenemos que dedicar los mayores esfuerzos a la detección y minimización de falacias. Para eso, tenemos que saber qué son, cómo son y dónde están.

En informática, tendemos mucho a cometer falacias desde el propio diseño del experimento, de diseño del dataset.

Cada vez que orientamos nuestro dataset y nuestro experimento a la consecución de buenos resultados, a la creación de un buen producto, a la publicación de un artículo científico, tenderemos a eliminar todo lo que nos resulte molesto. Por ejemplo, si hay una categoría de datos cuya inclusión consigue solamente una Test Acuracy del 65 % pero cuya exclusión nos permite llegar a un 97 %, tenderemos a eliminarla o a minimizarla. Esta forma de proceder es más habitual de lo que nos podemos imaginar. Pero es un error metodológico muy grave. Lo que hay que hacer en este caso es profundizar en el estudio de ese fenómeno, investigar las causas de ese comportamiento. Conocer mejor el ámbito en el que queremos aplicar un modelo Machine Learning o Deep Learning. Y gestionar adecuadamente lo que podamos descubrir.

Adaptar la realidad a nuestro modelo es el camino sencillo pero su poca rigurosidad es inasumible. En ningún caso debemos preparar una falacia desde el propio diseño del experimento, de forma parcialmente consciente o inconsciente incluso. Por eso debemos estar atentos en auto-detectarnos esos comportamientos.

Cuidado al buscar desesperadamente buenos resultados.

Siempre desconfíen de esos buenos resultados.

Cuestionen sus modelos SIEMPRE.

Pongan a prueba esos modelos con múltiples pruebas.

Intenten demostrar de todas las maneras posibles que están equivocados.

Si, después de todo eso, no pueden demostrar que están equivocados, es el primer paso: POR AHORA ESTÁN CORRECTOS.

Por ahora...

5.4. Las Dos Obsesiones del Método Científico

1. La repetibilidad.

Corroborada por la revisión por pares. Obtención de resultados consistentes al replicar un estudio con un conjunto diferente de datos, pero obtenidos siguiendo el mismo diseño experimental. En una revisión por pares, el experimento debe ser reproducible por cualquier persona en la comunidad científica y obtener prácticamente los mismos resultados.

2. La refutabilidad.

Las reglas y principios del método científico buscan minimizar la influencia de la subjetividad del científico en su trabajo. Se busca reforzar la validez de los datos obtenidos.

5.5. Refutabilidad

La refutabilidad (también llamada falsabilidad) se refiere a la capacidad de una teoría o hipótesis de ser sometida a pruebas que podrían demostrar que es falsa.

Este concepto implica que una afirmación científica debe ser formulada de manera que sea posible refutarla mediante evidencia empírica.

Si una teoría no puede ser sometida a pruebas que puedan contradecirla, no se considera científica según el criterio de refutabilidad de Karl Popper.

El falsacionismo es una corriente filosófica de la ciencia propuesta por el filósofo Karl Popper. Según esta perspectiva, una teoría científica debe ser falsable, es decir, debe ser posible refutarla mediante evidencia empírica. Popper argumentaba que una teoría solo puede considerarse científica si puede ser sometida a pruebas que, en principio, podrían demostrar que es falsa.

El inductivismo sostenía que las teorías se validan mediante la acumulación de evidencias que las confirman.

Por el contrario, el falsacionismo enfatiza la importancia de intentar refutar las teorías a través de experimentos y observaciones. Si una teoría sobrevive a las intentos de refutación y sigue siendo válida a pesar de las pruebas en su

contra, se considera provisionalmente aceptada, pero siempre sujeta a futuras pruebas que puedan falsificarla.

El falsacionismo propone que la ciencia avanza a través de la corrección de errores y la eliminación de teorías incorrectas, en lugar de confirmar teorías existentes. De esta manera, Popper buscaba establecer un criterio claro para distinguir entre afirmaciones científicas y no científicas, promoviendo un enfoque crítico y riguroso en la investigación científica.

En general, una proposición universal es falsable si existe al menos un enunciado lógicamente posible que se deduzca de ella, que pueda demostrarse falso mediante observación empírica. Si ni siquiera es posible imaginar un enunciado empíricamente comprobable que contradiga la proposición original, entonces tal proposición no será falsable.

Un ejemplo sencillo ayuda a entender el concepto. Para justificar la generalización 'todos los cisnes son blancos', según el verificacionismo tendríamos que buscar a todos los cisnes para comprobar que todos son blancos, algo prácticamente imposible. En cambio, según el falsacionismo, bastaría hacer lo contrario: buscar un cisne de cualquier otro color. Así, solo haría falta buscar un cisne diferente para refutar esa hipótesis, algo mucho más fácil.

Modus tollens es una forma de razonamiento lógico, una técnica de argumentación válida utilizada en lógica formal. Esta forma de argumentación se utiliza para inferir la falsedad de una afirmación a partir de la negación de su consecuencia.

Si de una afirmación P con condiciones iniciales se deduce lógicamente Q, es decir $P \rightarrow Q$, pero lo que se observa es $\neg Q$, entonces P es falsa, es decir $\neg P$.

En el ejemplo del cisne, dada la afirmación P 'todos los cisnes son blancos' y la condición inicial 'he aquí un cisne', se puede deducir Q 'este cisne es blanco'. Pero si es posible observar o al menos imaginar la observación $\neg Q$ 'este cisne no es blanco' (por ejemplo, es negro), la afirmación 'todos los cisnes son blancos' sería falsa, $\neg P$. Esto comprueba que la afirmación inicial es falsable.

El verificacionismo es una doctrina filosófica que sostiene que una afirmación o enunciado solo tiene significado si puede ser verificado mediante la observación o la

experiencia empírica. Según esta perspectiva, las afirmaciones que no pueden ser comprobadas de esta manera, ya sea directa o indirectamente, carecen de significado y son consideradas como vacías o sin sentido.

Para justificar la generalización 'todos los cisnes son blancos', de acuerdo con el verificacionismo, tendríamos que buscar todos los cisnes para verificar si todos son blancos, algo prácticamente imposible.

Por otro lado, de acuerdo con el falsacionismo, bastaría encontrar un cisne de cualquier otro color. Así, solo sería necesario encontrar un cisne diferente para falsificar esa hipótesis, algo mucho más fácil.

5.6. Refutabilidad en Computación

En el conjunto de experimentos de aplicación de Machine Learning y Deep Learning es difícil encontrar metodologías orientadas expresamente a medir la refutabilidad del experimento.

En muchos casos, se da prioridad a la precisión y al rendimiento del modelo sobre su capacidad para ser desafiado o refutado.

De momento sólo queremos indicar que la aplicación correcta de estos modelos en áreas sensibles desde la biosanitaria hasta la aplicación en el ámbito de la justicia, pasan inevitablemente por una medición extensa de la refutabilidad de los modelos. Hasta que los ingenieros no hagamos esto, los expertos en estas áreas no nos tomarán en serio. Esto es muy duro pero es así. Extenderemos este asunto a lo largo de los siguientes capítulos.

5.7. Replicabilidad

Es la capacidad de un estudio o experimento para ser repetido por otros investigadores bajo condiciones similares y obtener resultados consistentes.

En otras palabras, un estudio es considerado replicable si otros científicos pueden llevar a cabo el mismo experimento y obtener los mismos resultados o resultados similares.

La replicabilidad es esencial para la validación y la confiabilidad de los hallazgos científicos. Si los resultados de un estudio no pueden ser replicados por otros investigadores, esto plantea serias dudas sobre la fiabilidad y validez de esos resultados. Por lo tanto, la replicabilidad es un criterio importante para determinar la solidez de la evidencia científica.

Existen varios factores que pueden afectar la replicabilidad de un experimento:

1. Diseño del experimento. La calidad del diseño experimental y la claridad de los procedimientos pueden influir en la replicabilidad. Un diseño experimental claro y bien documentado facilita la reproducción del estudio por otros investigadores.

2. Tamaño de la muestra. El tamaño de la muestra utilizado en un estudio puede afectar la replicabilidad. Los estudios con muestras pequeñas pueden producir resultados menos consistentes debido a la variabilidad aleatoria, mientras que los estudios con muestras más grandes tienden a ser más robustos y replicables.

3. Precisión y Descripción Detallada de los Procedimientos y Metodologías. Si los procedimientos y metodologías no se describen claramente o no se reproducen con precisión, puede ser difícil para otros investigadores replicar los resultados.

4. Transparencia y Compartición. La transparencia en la presentación de datos, métodos y análisis estadísticos contribuye a la replicabilidad. Compartir datos, código y materiales experimentales permite a otros investigadores verificar y replicar los resultados.

5. Factores contextuales. Algunos estudios pueden verse afectados por factores contextuales o condiciones específicas del entorno en el que se llevó a cabo el estudio original. Estos factores pueden dificultar la replicabilidad en entornos diferentes.

La replicabilidad es fundamental para garantizar la fiabilidad y validez de los resultados.

5.8. Replicabilidad en Deep Learning

Dos entrenamientos de un mismo modelo, de una misma arquitectura, nunca tendrán el mismo resultado. Ni los mismos ratios ni las mismas clasificaciones. Es

decir, en un entrenamiento, una imagen puede ser correctamente clasificada por un modelo; y en otro entrenamiento, no.

No obstante, un modelo robusto siempre proporcionará ratios en el mismo rango, por ejemplo una Test Accuracy de un $95\,\% \pm 0{,}2\,\%$.

La cuestión a valorar es si esa horquilla es suficiente para el objeto de aplicación del modelo.

Algunas de las razones por las que en Deep Learning no es posible una repetibilidad absoluta son las siguientes:

Inicialización aleatoria. Muchos modelos de aprendizaje profundo utilizan inicializaciones de pesos aleatorios. Esto significa que cada vez que se inicia un entrenamiento, los pesos del modelo se inicializan de manera diferente, lo que puede llevar a resultados ligeramente diferentes incluso con los mismos datos y hiperparámetros.

Optimización no determinista. Los algoritmos de optimización utilizados para entrenar modelos de aprendizaje profundo, como el Descenso de Gradiente Estocástico (Stocastic Gradient Descent, SGD), pueden ser no deterministas. Esto significa que, incluso con los mismos datos y hiperparámetros, el proceso de optimización puede converger hacia diferentes mínimos locales en diferentes ejecuciones.

Condiciones del entorno y hardware. El entorno de ejecución, incluidos los recursos de hardware como la GPU y el sistema operativo, puede afectar el rendimiento y la reproducibilidad del entrenamiento del modelo.

Un asunto importante: la gran ventaja de los modelos Deep Learning es su capacidad de generalizar. Es decir, su capacidad de abstraer el conocimiento de sus datos de entrenamiento a datos nuevos que no ha visto nunca.

La generalización implica necesariamente la asunción de un margen de error.

Si precisamos una repetibilidad absoluta o de un error del $0\,\%$, no debemos utilizar modelos Deep Learning. No hay más. Así de simple.

5.9. Algunos Consejos para Aproximar estos Algoritmos al Método Científico

Antes hemos apuntado qué tenemos que hacer como ingenieros para que los expertos en las ramas del conocimiento donde aplicamos Machine Learning y Deep Learning nos tomen en serio. Es una afirmación muy dura, somos conscientes.

Veamos alguna de las críticas de expertos en el ámbito biosanitario en cuanto a la aplicación de estos algoritmos en su área[3-14].

Los autores estamos de acuerdo con todas ellas.

Empezaremos por las más leves:

- Es necesario llevar a cabo investigaciones más estandarizadas y más alineadas con los procedimientos de los estudios clínicos.

- Es necesario diseñar algoritmos más adaptados a la genómica. Que no estén tan enfocados en lograr la mayor precisión, sino en integrar eficientemente las peculiaridades de la genómica.

- Es fundamental optimizar la mejora en la generalización de los modelos, especialmente teniendo en cuenta la alta variabilidad del conjunto de entrenamiento y la importancia de los casos menos usuales.

- Existen limitaciones intrínsecas al Deep Learning que hacen difícil sustituir los métodos tradicionales de investigación y detección.

- El Deep Learning no debe ser interpretado ni sobreestimado, ni en la academia ni en la industria de la IA. De hecho, tiene muchos problemas técnicos por resolver debido a su naturaleza.

Y ahora las más duras:

- Muchos estudios cuestionan el desempeño real y solicitan más transparencia y honestidad en los resultados obtenidos.

- Muchos alertan sobre graves fallos de aplicación.

- Muchos de ellos los excluyen directamente para su aplicación clínica.

Es un fenómeno curioso. Mientras los expertos en computación se enorgullecen de los avances, los expertos en el campo biosanitario son mucho más escépticos.

Queremos en este momento lanzar una pregunta: ¿Ustedes se dejarían operar a corazón abierto por un robot gestionado íntegramente por 'inteligencia artificial'? Sin intervención humana, ni siquiera supervisando. Sólo ustedes y el robot.

Si la respuesta es NO, bueno, evidentemente, aún queda mucho por hacer para implementar desarrollos serios en el ámbito. Ahora estamos más cerca de lo que opinan tantos expertos en el ámbito biosanitario.

¿Qué podemos hacer para que los 'clientes finales', los expertos en las áreas de aplicación tomen en serio los algoritmos Deep Learning en sus áreas?

1. Vamos intentar verificar si no cometemos falacias consciente o inconscientemente.

2. Prestaremos mucha atención a la repetibilidad y a la refutabilidad. Aseguraremos que nuestro experimento sea repetible y lo someteremos a los estándares más exigentes en cuanto a refutabilidad. Para esto nos apoyaremos en las metodologías estándar de los ámbitos de aplicación.

En informática, el control de la falsificabilidad es especialmente importante. Desde el momento en que dirigimos nuestro experimento para obtener buenos resultados computacionales, siempre estamos sesgando el conjunto de datos.

Tendemos a generar un conjunto de datos muy personalizado, muy orientado a lo que queremos lograr. Y LO HAREMOS PRÁCTICAMENTE SIN DARNOS CUENTA. Lo veremos un poquito más en el siguiente capítulo.

Podemos incluso obtener buenos resultados (AUC, Matriz de Confusión...). Pero eso no quiere decir que sean realmente válidos.

Debemos dedicar mucho tiempo y esfuerzo a la detección de sesgos en nuestra experimentación. Aunque eso suponga tirar abajo modelos que aparentemente funcionaban bien.

Y los ingenieros en computación serán lo más multidisciplinares posible. Es imperativo aprender del ámbito en el que se trabaja. Si no hacemos esto, el resultado NUNCA va a ser el mismo.

Nuestro objetivo como ingenieros es poner nuestras herramientas AL SERVICIO de un fin específico. Nuestras herramientas son solo un medio, no son ese fin. Por favor, recuerden esto siempre...

Cuando veas a un hombre bueno, trata de imitarlo. Cuando veas a uno malo, reflexiona sobre ti mismo

— Confucio, *toda su vida intentó persuadir a los gobernantes para que siguieran sus enseñanzas, pero nunca logró nada más que un puesto público de bajo escalafón. Desde hace 2.500 años es uno de los sabios más influyentes en la humanidad.*

6 Los Sesgos, Ese Gran Problema...

Índice

6.1. Los Algoritmos Machine Learning No Entienden al Ser Humano

Esto es un hecho. Cuanto más se acercan a aspectos intrínsecamente humanos, más fallan. No son capaces de parametrizar sentimientos complejos, matices del lenguaje, expresiones faciales profundas, ricas y sutiles...

Pongamos un ejemplo muy simple: estos modelos tienen serios problemas de

detección de cinismo, ironía, etc. No obstante, a veces nos sucede también a los humanos, aunque se nos da algo mejor.

La consecuencia inmediata de estas dificultades es una complejidad alta en la detección de toxicidad en redes sociales, por ejemplo.

6.2. Los Algoritmos No Tienen Sesgos. Los Seres Humanos, Sí

Hemos visto la importancia de la subjetividad y los prejuicios en el método científico. Esta importancia no es casual.

La capacidad del ser humano para generar sesgos, prejuicios, es tendente al infinito. Todos tenemos sesgos, todos tenemos prejuicios.

Esos prejuicios, esas falacias, esos estereotipos, influyen y contaminan nuestros datasets.

Los algoritmos no tienen capacidad para tener prejuicios. Son formulaciones matemáticas, no pueden generar sesgos por iniciativa propia.

Los modelos Machine Learning y Deep Learning sólo son algoritmos entrenados con datos. Con datos proporcionados por humanos. Aquí empieza el problema: esos datos están sesgados por nosotros, los humanos.

Por lo tanto:

Los prejuicios en estos modelos son los prejuicios humanos.

Mientras no aceptemos esto, no podremos solucionar nada.

Así de simple.

6.3. ¿Somos Conscientes de Nuestros Propios Sesgos, de Nuestros Prejuicios?

Creemos que no.

Llevamos unos 2.500 años intentando ser imparciales, ser objetivos. Y aún hoy no lo hemos conseguido. Tal vez porque nuestra subjetividad es inconsciente.

Hemos visto en múltiples ocasiones cómo una 'inteligencia artificial' ha sido desconectada, eliminada, porque 'se ha vuelto' racista, misógina, xenófoba, homófoba, etc. y que incluso ha llegado a la lógica conclusión de que hay que destruir a la humanidad. Esto nos puede parecer hilarante. Pero detrás de todo esto simplemente tenemos que dedicarnos a leer contenidos en redes sociales. Y entenderemos por qué suceden estos fenómenos.

Imagínense un extraterrestre que quiere aprender qué somos los humanos. Y lee todo lo que hemos escrito en internet. Y lee todo lo existente en redes sociales para aprender de primera mano cómo pensamos, cómo nos relacionamos, como reaccionamos... Cómo somos. ¿Qué pensaría ese extraterrestre?

A veces exigimos a estos algoritmos lo que no nos exigimos a nosotros mismos.

En Internet hay muchísimo vídeos de experimentos variopintos para la detección de sesgos en el ser humano. Invitamos al lector a que investigue.

Hay vídeos que demuestran que no nos comportamos igual ante una persona que demanda ayuda y aparenta una clase social alta frente a una persona que demanda ayuda y está en riesgo de exclusión.

Hay vídeos que demuestran que es más fácil para una mujer captar el interés si va vestida de una determinada manera que si va vestida de una manera descuidada.

Hay vídeos que demuestran que una persona blanca consigue ayuda más fácilmente que una persona negra.

Hay vídeos muy duros. Y podemos aprender mucho de ellos. Ninguna persona estamos libres de prejuicios. Y todos esos prejuicios de los que no somos conscientes, los compartimos, los transmitimos y no nos damos cuenta. Porque son inconscientes.

Suele ser útil analizar lo primero que nos viene a la cabeza. Esa primera idea es la que suele incluir libremente esos sesgos inconscientes. Inmediatamente después, la filtramos, la adaptamos a lo correcto y tapamos esos sesgos que influyen nuestro comportamiento sin darnos cuenta. Pero los sesgos siguen ahí.

Por ejemplo:

- Visualice una persona de éxito. ¿Cómo es esa primera imagen que ha visualizado? ¿Es un hombre o es una mujer? ¿De qué raza? La respuesta es sólo para usted, nadie se va a enterar.

- Se le ha estropeado el coche, lo aparca al lado de una acera y busca ayuda. Se acercan un hombre y una mujer. ¿A quién pide ayuda de los dos?

Contaremos una anécdota real (y anónima):

'Al aparcar hice una hendidura en el neumático de mi coche. No sabía si esa hendidura era grave o no suponía un problema inmediato. En ese momento pasaron a mi lado un hombre y una mujer caminando juntos. Me dirigí al hombre. No se ni por qué, me salió solo. El hombre, visiblemente incómodo no sabía qué decirme y le preguntó a su mujer. Su mujer sí pudo ayudarme, aunque estaba lógicamente molesta conmigo. Me sentí ridícula, me sentí mal, fue una estupidez ni siquiera considerar que cualquiera de los dos estaba cualificado para ayudarme. Me dirigí directamente al hombre sin valorar ni siquiera la posibilidad de que esa mujer también podía. Y soy una mujer.

Esta situación me dio una lección de vida que no olvidaré jamás. Ahí actuó un prejuicio. Los hombres son los que saben de mecánica, de coches. Las mujeres no.

Fui una estúpida y no me cuesta reconocerlo. Es la única forma de aprender.'

Otra anécdota (real y anónima también):

'Muchos de mis grandes amigos son de otros países y de otras razas, de otros colores de piel, llamémoslo como queramos. Charlando un día de racismo, me reconocieron que, cuando se montan en un tren, un metro, un autobús, están acostumbrados a que la gente española blanca evite sentarse con ellos. Siempre tienden a sentarse al lado de otro blanco. Lo contaban sin enfado, sólo resignación. Lo tienen perfectamente asumido y naturalizado. Es más, lo justifican, es que no se dan ni cuenta, no pasa nada, no lo hacen adrede, no son conscientes, es normal.

Sentí mucho dolor al escuchar todo eso. Sentí pena por mis amigos, por todos los de un color de piel diferente. Por ese sentimiento de resignación. Sentí mucha pena. Y mucha vergüenza.

Nunca me hubiese imaginado que eso era posible. Tiendo a ser autocrítica, así que lo siguiente que me planteé es si yo, sin darme cuenta, lo había hecho también.

No lo sé, no soy consciente pero tengo que pensar que sí, que lo he hecho. Desde ese día, cada vez que me suba a un tren, un metro, un autobús, me siento deliberadamente al lado de una persona cuyo color de piel es diferente al mio. Para que, por al menos una vez, no sienta lo que sienten mis amigos todos los días de su vida.'

El magnífico ensayo de Federico Navarrete, *'México Racista. Una Denuncia'*[15], en su Sección *'Las muñecas de colores'* describe un vídeo realizado El Consejo Nacional para Prevenir la Discriminación y la productora 11-11 Cambio Social. En este vídeo se ve cómo inevitablemente unos niños terminan asociando todas las virtudes planteadas a un muñeco blanco y todos los defectos planteados a un niño negro. Cuando se le pregunta a qué muñeco se parecen más, las respuestas tienden al surrealismo. Quieren parecerse a toda costa al muñeco blanco, aunque los niños con la piel más oscura tienen serias dificultades para justificarlo. Qué sufrimiento, pobres niños...

Toda una lección de vida. Desde que descubrimos esta dinámica, cuesta sinceramente quitarnos de la cabeza a esos niños. ¿Qué habrán sentido después de ese vídeo? ¿Qué habrán pensado? ¿Y sus padres? Cuánto dolor tiene que haber detrás de todo esto...

Por supuesto, ni esos niños ni esos padres son racistas. Esto es el reflejo de una sociedad que nos ha inculcado a fuego en lo más profundo de nuestra mente esas formas de pensar, esas formas de sentir. De sentirnos. Y esto no es exclusivo de un país concreto. Estamos seguros de que en todos los países se repite este mismo fenómeno.

Estos son los sesgos.

Y estos sesgos llegan muy lejos.

Si analizamos en detalle registros médicos en Estados Unidos, veremos que esos sesgos asociados a la raza están ahí, en esos registros médicos. El ejemplo típico, el ejemplo facilón es Estados Unidos, pero sucede en todos los países. Debemos hacer constar este punto. Ningún país se libra.

Volvamos al manido ejemplo de Estados Unidos: La atención médica proporcionada a las personas blancas es mejor que la ofrecida a las personas negras. Y esto es cuantificable y está reflejado en varios estudios[16-19].

Sesgos en la atención cardiovascular a mujeres frente a hombres. A las mujeres se les diagnostican los infartos más tarde y peor. Porque los infartos son típicos de los hombres y 'las mujeres podemos ser un poco histéricas' (léase con tono irónico, por favor)[20-22].

Si nos auto-analizamos y analizamos la sociedad en la que vivimos, descubriremos múltiples ejemplos de prejuicios, a todos los niveles.

Recuerden: quédense con su primer pensamiento, su primera imagen mental, su primera reacción antes de ser filtrada. Ahí están los sesgos. Hagan este ejercicio de auto conocimiento, es profundamente enriquecedor.

Hay una dificultad adicional en el control de los sesgos. Los sesgos sufren modificaciones en función de muy diversas realidades. Los sesgos evolucionan, cambian, nacen nuevos sesgos y tal vez desaparecen otros. La realidad socio política, económica, factores religiosos, culturales, ámbitos geográficos, incluso las modas, todo esto puede modificar nuestros sesgos. Al igual que puede modificar nuestra forma de pensar, de actuar o de sentir.

Los sesgos son inherentes al ser humano por lo que están influenciados por todos y cada uno de los factores influyentes en el propio ser humano, de alguna manera.

Disponen de muchísima bibliografía.

Lean mucho sobre los sesgos, sobre los prejuicios. Si se interesan por el tema y se documentan, les garantizamos que van a aprender mucho de nuestra sociedad. Y, sobre todo, de uno mismo.

6.4. El Reflejo de los Sesgos Humanos en Machine Learning

Somos los seres humanos los que etiquetamos los datos.

Y en ese etiquetado van todos nuestros sesgos. Algunos más controlados y otros menos controlados. Y esto dependerá de cada etiquetador.

Y en los propios datos existentes también van nuestros sesgos.

Vamos a aplicar un algoritmo Deep Learning a los registros médicos de Estados Unidos. Ese sesgo inherente a los propios datos, donde las personas blancas reciben mejor asistencia que las negras, estará presente a lo largo de todo el entrenamiento del modelo. Los registros de personas blancas son abundantes, completos y correctos. Y los de las personas negras son pocos, mal hechos y con un diagnóstico incorrecto en consecuencia.

Y el modelo aprenderá ese sesgo y lo aplicará ante nuevos datos de entrada. Esto es así y no tiene solución.

Mismo caso con la detección de infartos en mujeres. La desatención a las mujeres tiene que como consecuencia que en el histórico de registros médicos haya poca información de enfermedades cardiovasculares incipientes en mujeres (no hubo usualmente una atención en las primeras fases), o los datos son incorrectos por un mal diagnóstico que sucedió más frecuentemente de lo deseado.

Esas deficiencias en el diagnóstico cuando el paciente es una mujer no se pueden eliminar, son los datos médicos de muchos años. Cualquier modelo que entrene con esos datos, aprenderá y repetirá. La culpa volverá a ser nuestra.

Pobres mujeres negras...

Se requerirán nuevos millones de registros médicos equitativos para conseguir un modelo equitativo. Pero la culpa no es del modelo. La culpa es nuestra.

Supongamos un modelo entrenado para realizar selecciones de personal para puestos directivos en grandes empresas. Sólo tenemos que ver el porcentaje de mujeres directivas vs. hombres directivos (piensen de nuevo en su primera imagen de 'persona de éxito').

No podemos pretender que un modelo Deep Learning seleccione mujeres pudiendo seleccionar hombres, cuando las estadísticas, REALES, son escandalosas.

En el mejor de los casos, el modelo seleccionará mujeres en la misma proporción que la cruda realidad. Por lo tanto, evidentemente, cualquier modelo será sexista porque la realidad es sexista. La culpa no es del modelo. La culpa es nuestra.

El método científico lleva desarrollándose desde hace unos 2.500 años. Desde el principio, su gran objetivo ha sido minimizar el impacto de la subjetividad, de los prejuicios, de los sesgos. Aún hoy seguimos luchando para conseguirlo. Aún no hemos conseguido tener los sesgos bajo control, siquiera.

Este asunto es tan serio que incluso se está aprovechando la ventaja de que los modelos Machine Learning y Deep Learning no pueden tener sesgos, precisamente para detectarlos. Y precisamente en datos biomédicos. Por lo tanto, el asunto es grave.

Por lo tanto, una vez más, no pidamos a unos algoritmos lo que nosotros no somos capaces de hacer. De hecho, una de las ventajas de estos algoritmos es que nos ponen un espejo delante para que nos veamos reflejados. Estos algoritmos denuncian nuestros propios sesgos. Y, por lo tanto, nos ofrecen una oportunidad para aprender. Y ser mejores.

6.5. Un Par de Agravantes al Problema de los Sesgos

Queremos señalar dos fenómenos (por supuesto, humanos) que pueden agravar el problema de los sesgos. Y, por lo tanto, estropear aún más el dataset para entrenar modelos Deep Learning.

6.5.1. El Poder de la Influencia Social

Los humanos somos animales gregarios. Pertenecemos a una manada, no lo podemos evitar. Lo llevamos dentro desde incluso antes de ser realmente humanos.

El fuerte sentimiento de pertenencia a una manada, a una sociedad, hace que nuestro comportamiento, nuestro pensamiento y nuestro sentimiento esté altamente influenciado por las acciones, opiniones y presiones de esa sociedad, de esa manada. Este fenómeno ha sido estudiado en diversas disciplinas, como la psicología social, la sociología y la antropología, y se manifiesta en una variedad de contextos sociales.

Algunos de los aspectos clave del fenómeno de la influencia social incluyen:

1. Conformidad. La conformidad es el cambio en el comportamiento o las creencias de una persona para ajustarse a las normas sociales del grupo. Esto puede

ocurrir tanto en situaciones públicas como privadas y puede ser influenciado por factores como el tamaño del grupo, la cohesión del grupo y el grado de unanimidad en la opinión del grupo.

2. Obediencia. La obediencia se refiere a la disposición de una persona a seguir las órdenes o instrucciones de una autoridad, incluso si esas órdenes entran en conflicto con sus propios valores o creencias. Los experimentos clásicos de Stanley Milgram sobre la obediencia a la autoridad ilustran este fenómeno, mostrando cómo las personas pueden obedecer órdenes para infligir dolor a otros cuando se les ordena hacerlo por una figura de autoridad[23].

3. Influencia de los roles sociales. Los roles sociales, o las expectativas y comportamientos asociados con una posición particular en la sociedad, también pueden influir en el comportamiento de las personas. Esto puede incluir roles basados en la edad, el género, la ocupación u otras características sociales.

4. Polarización del grupo. La polarización del grupo se produce cuando las opiniones y actitudes de un grupo se vuelven más extremas después de la discusión grupal. Esto puede ocurrir debido a la tendencia de las personas a buscar validación social y a adoptar posiciones más extremas para ser coherentes con la identidad del grupo.

¿En qué se traduce el poder de la influencia social? En que haremos todo lo posible, consciente e inconscientemente, para que el grupo no nos rechace como miembros.

En Internet hay vídeos francamente divertidos sobre personas que se montan en un ascensor y se colocan en la misma posición que el resto de personas que ya están dentro. Es un ejemplo muy simple pero la gran lectura es que la influencia social ejerce un poder fortísimo sobre nosotros.

Las personas que están en el ascensor no son un grupo importante, no es nuestra familia, nuestro grupo de amigos... Y sin embargo, cuando nos subamos en un ascensor, tenderemos a colocarnos en la misma posición que el resto, aunque esa posición sea extraña.

Somos capaces de adaptarnos al grupo aunque ese grupo no tenga ninguna importancia en nuestra vida. Imaginen lo que somos capaces de hacer si el grupo es importante para nosotros.

Esta influencia social, normalmente tiende a potenciar nuestros sesgos, rara vez a minimizarlos. La retroalimentación entre los individuos muy frecuentemente genera polarización. Y normalmente para mal. Además, cuanto más homogéneo sea el grupo, más similares sus miembros, más exclusión ante lo diferente, con altísima probabilidad.

Y ya que hablamos de polarización...

6.5.2. El anonimato en redes sociales

Aquí vamos a elevar el tono. No vamos a andar con paños calientes o un lenguaje suavizado para ser políticamente correctos. No vamos a maquillar el problema con palabras a medias tintas.

Cuántos modelos han tenido que ser re-entrenados porque se convirtieron en monstruos después de aprender de las redes sociales...

Es bien sabido que, sobre todo en determinadas redes sociales, la cantidad de basura (disculpen, no hay otra palabra) que se escribe es totalmente vergonzosa. ¿En serio pretendemos obtener un modelo maravilloso al que le damos esta basura para aprender? ¿En serio?

La culpa no es del modelo, la culpa es nuestra.

El debate filosófico aquí es riquísimo y eterno, pero el objetivo de este libro no es profundizar en la bondad inherente al ser humano.

Vamos a hablar simplemente de hechos fácilmente comprobables.

El fenómeno de los haters.

Ha muerto gente por su culpa. Esto sucede prácticamente en su totalidad en redes sociales. No hay haters volando libres en el mundo real. ¿Por qué?

Muy sencillo: por cobardía. Así de simple. Cobardía en dos vertientes:

- Primera vertiente de cobardía: EL ANONIMATO. Se ocultan en el anonimato que les proporcionan las redes sociales con escandalosa facilidad. Y así pueden

escribir lo que les dé la gana. Y destruir a quien les dé la gana también. Y NADIE SABE QUE SON ELLOS. No se atreven a identificarse porque no se quieren responsabilizar de sus opiniones.

- Segunda vertiente de cobardía: EL PODER DEL GRUPO. Los haters actúan en manada a ser posible, como todos los abusadores. Es muy difícil que un abusador actúe solo. En grupo se resguardan y se protegen. Porque todos son unos cobardes y el grupo minimiza esa cobardía. Las redes sociales son el escenario ideal para que estos matones actúen, en banda a ser posible, porque solos no se atreven.

Así de simple y así de crudo. Y este atajo de cobardes se dedica a arruinar vidas por mera diversión, por sentirse poderosos en su inmundicia.

Las redes sociales, con su apología del anonimato, fomentan estos comportamientos.

Por nuestra parte, nuestras opiniones escritas son exactamente las mismas con anonimato o sin él. Porque estamos orgullosos de lo que opinamos.

¿Los haters lo están? ¿Por qué no salen de ese anonimato y dan la cara?

Este anonimato influye tantísimo en la potenciación de los instintos más bizarros del ser humano que, una vez más, tenemos un dataset lleno de datos absurdos, horribles y ajenos a la realidad. Por lo tanto, un dataset inválido.

Un algoritmo, un extraterrestre, un folio en blanco, un niño pequeño: TODOS TERMINARÁN SIENDO PARECIDOS SI APRENDEN CON ESO.

La culpa no es del algoritmo. La culpa es nuestra.

6.6. El Resultado de Todo Esto

Por lo tanto, si juntamos:

- Todos los sesgos perfectamente plasmados en Internet.

- Las informaciones erróneas y/o maliciosas.

- El poder de la influencia social que potencia todo lo anterior.

- Y el anonimato, que eleva a la enésima potencia todo lo negativo en redes sociales.

Tenemos:

EL DATASET PARA ENTRENAR UN ALGORITMO.

No pidamos a un algoritmo lo que nosotros somos incapaces de hacer.

6.7. Correlaciones Espurias

Las correlaciones espurias son asociaciones aparentes entre dos variables que en realidad no tienen una relación causal directa. Estas correlaciones pueden surgir cuando dos variables están asociadas con una tercera variable oculta o cuando ocurre una coincidencia fortuita.

Por ejemplo, supongamos que se observa una correlación positiva entre el consumo de helado y el número de ahogamientos en una piscina. Podría ser tentador concluir que el consumo de helado aumenta el riesgo de ahogamiento. Sin embargo, la correlación en este caso es espuria, ya que ambas variables están influenciadas por una tercera variable oculta: la temperatura. En los días calurosos, es más probable que las personas consuman helado y también es más probable que naden en piscinas, aumentando así el riesgo de ahogamiento. Por lo tanto, la temperatura es la variable que está causando tanto el aumento en el consumo de helado como el aumento en los ahogamientos, y no hay una relación causal directa entre el consumo de helado y los ahogamientos.

Es importante tener en cuenta las correlaciones espurias al analizar datos y hacer inferencias, ya que pueden llevar a conclusiones erróneas si no se consideran adecuadamente las variables ocultas o las coincidencias fortuitas. Para evitar este tipo de errores, es fundamental realizar un análisis cuidadoso y considerar todas las posibles explicaciones alternativas para las correlaciones observadas.

Queremos recomendar una página web fantástica con muchos ejemplos de correlaciones espurias. Por favor, si tienen la oportunidad, visítenla, merece la pena: https://tylervigen.com/spurious-correlations. Es una página fantástica.

Veamos algunos ejemplos de correlaciones espurias:

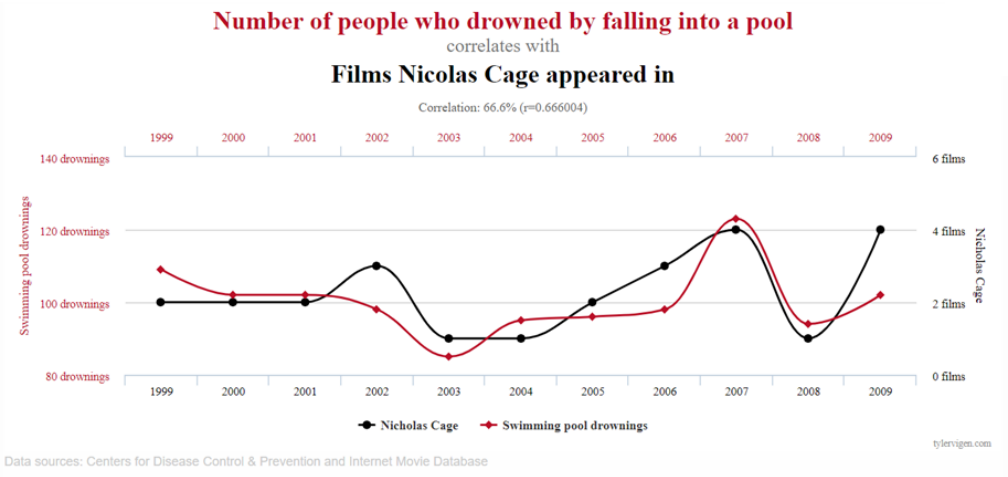


Figura 6.1: Un Ejemplo de Correlación Espuria. Extraído de https://tylervigen.com/spurious-correlations.

En esta gráfica vemos la evolución anual paralela entre el número de personas ahogadas en una piscina y el número de películas en las que actuó Nicolas Cage ese año.

Son dos variables cuyas tendencias son similares. La mera visualización entre ambas variables nos lleva inmediatamente a pensar que están relacionadas, que hay una causalidad entre una y la otra. No sabemos qué relación tienen, qué causalidad las determina ni cuál depende de cuál. Pero están relacionadas. Esta es nuestra conclusión a la vista de esta gráfica.

El sentido común nos lleva a determinar que no hay relación entre ambas. El hecho de que coincidan sus evoluciones es una mera casualidad.

Los algoritmos no tienen sentido común, así que, sin más datos, concluirán que evidentemente el número de ahogados en piscina depende del número de películas en las que actúa Nicolas Cage, o viceversa.

Veamos otro ejemplo:

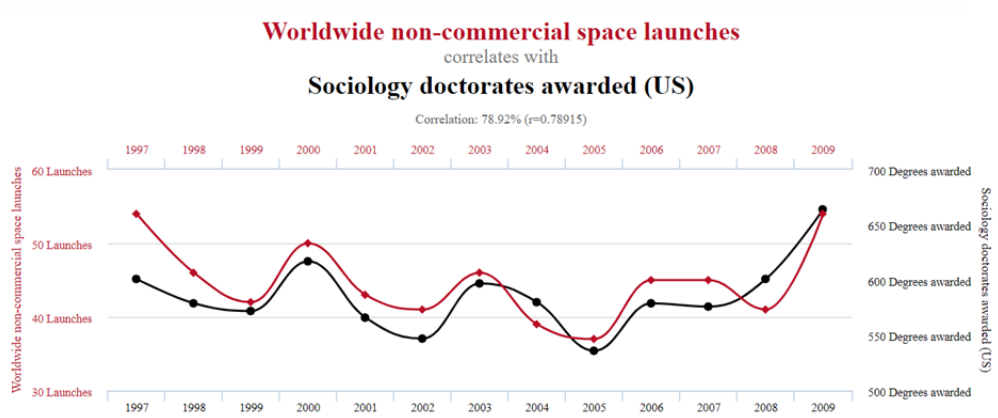


Figura 6.2: Otro Ejemplo de Correlación Espuria. Extraído de https://tylervigen.com/spurious-correlations.

Esta gráfica es aún más divertida porque el grado de similitud entre ambas curvas es aún mayor. Aquí podemos ver el número anual de lanzamientos espaciales no comerciales a nivel mundial con el número anual de Doctorados en Sociología en Estados Unidos.

¡Las evoluciones de estas dos variables van de la mano! ¿Hay causalidad entre ambas? Sentido común: no. Estos lanzamientos especiales no dependen de la cantidad de Doctorados en Sociología. Y al contrario tampoco.

Pero lo parece tanto...

Estos ejemplos son muy divertidos. Vamos a inventar un caso menos divertido:

Supongamos que realmente existe una correlación espuria entre el comportamiento de un sistema inmunológico frente a un patógeno y un alto nivel de glucosa en sangre. Y nosotros no sabemos que existe esa correlación espuria. Podríamos fácilmente concluir que los individuos con niveles altos de glucosa en sangre, se comportan de esta manera frente a este patógeno. Y ESO NO ES CIERTO. Todas las medidas tomadas a continuación serían incorrectas y probablemente peligrosas.

En este ejemplo es fácil pensar que puede que exista una relación desconocida entre la relación entre el alto nivel de glucosa y el comportamiento inmunológico. Es muy cierto. Puede ser, puede existir una relación directa o indirecta. En ese caso, no estaríamos hablando de una correlación espuria.

Antes de llegar a una conclusión u otra (correlación espuria vs. relación desconocida), es preciso asegurarse muy bien de qué estamos hablando. Y si no es posible, trabajar con la hipótesis más conservadora, o aquella cuyos resultados sean menos peligrosos, o aquella más probable... La opción que sea mejor en cada caso de estudio.

Aquí simplemente hemos hecho el supuesto de que hemos demostrado fehacientemente que hay una correlación espuria entre ambas variables.

Es preciso verificar siempre que las correlaciones no son espurias. Si se puede verificar de múltiples maneras, mejor que mejor.

6.8. Consejos para la Minimización de Sesgos

Es una tarea extremadamente compleja.

Un primer paso muy enriquecedor es desarrollar una profunda tarea de auto análisis a nivel individual para identificar y controlar nuestros propios sesgos. Un ejercicio de autocrítica constructivo y exhaustivo, realizado con humildad y apertura al cambio interior.

Muchos de nuestros sesgos se producen por un desconocimiento de la realidad asociada a esos sesgos. Por lo tanto, es muy útil investigar y profundizar en el conocimiento de todo aquello que nos produzca sesgos. Por ejemplo, si en ese auto-análisis detectamos que tenemos prejuicios con una determinada raza, nacionalidad, religión, condición sexual, etc. es muy necesario aprender de esa raza, nacionalidad, religión, condición sexual, etc. Documentarnos con datos objetivos medibles, acercarnos a esa realidad que nos ha generado cierta distancia, desconfianza o incluso cierto rechazo.

Un mayor conocimiento nos lleva inevitablemente a una mayor cercanía. Muy probablemente encontraremos puntos en común. Y muy probablemente los sesgos se reducirán, al menos en parte.

Puede que no se reduzcan, puede incluso que aumenten. Pero en cualquier caso, tendremos un mayor auto conocimiento y esos sesgos no serán tan inconscientes.

Lo más habitual es que se reduzcan, en cualquier caso.

Este proceso de crecimiento intelectual nos llevará también a una mayor capacidad de detección inmediata de sesgos propios y ajenos. Otra ventaja añadida.

Estamos hablando de cómo minimizar problemas inherentes a la naturaleza de algoritmos Machine Learning y Deep Learning. Sin embargo, estamos hablando en términos psicológicos, sociológicos, etc. Son dos enfoques muy diferentes.

Realmente no lo son. Estos algoritmos pretenden imitar ciertos procesos intelectuales humanos. Es lógico por tanto que la optimización de la aplicación de estos algoritmos provenga del análisis de esos procesos intelectuales.

Es fundamental tomar en consideración lo expuesto en esta sección para conseguir minimizar los sesgos inherentes a los datos con los que entrenan estos modelos.

Supongamos un algoritmo que tiene que detectar vídeos violentos en una red social. Actualmente (año 2024) lo habitual es etiquetar cada vídeo del Training Set, fotograma por fotograma, como 'violento' o 'no violento'. Más o menos, todos tenemos claro lo que es violento y lo que no. Pero la dificultad comienza con la siguiente pregunta: ¿Dónde está el límite, la línea roja que separa un contenido violento de uno que realmente no lo es?

¿Ese límite es el mismo entre, por ejemplo, una monja y un narcotraficante? ¿La justificación de cierta violencia sutil puede llevarnos a no considerarlo violencia? ¿La costumbre a altos niveles de violencia puede llevarnos a no considerar violencia ciertos comportamientos no tan escandalosos?

Hace 100 años un cachete a un niño en el culete era considero simplemente educación. Hoy es violencia. El concepto de violencia cambia, al igual que los sesgos.

¿Les parece violencia un burka? ¿A un talibán u otro tipo de extremista le parece violencia? ¿Pensamos igual? Sin duda: no, no pensamos igual.

Si transferimos esto al etiquetado de datos, pueden pasar muchas cosas.

Es un asunto muy complejo y sin solución buena. Sólo podemos apuntar ciertas formas de trabajar:

- Organizar un dataset con miles de datos. Cuantos más datos, mejor.

- Que contengan todas las casuísticas de forma suficiente. Cuanto más variado sea el dataset, mejor. Es muy conveniente incluir datos de aquellas casuísticas que,

aunque no son objeto de estudio, existen. Por ejemplo, un clasificador que distinga coches de camiones en vídeos. Será interesante incluir imágenes de motocicletas, furgonetas, caravanas, aviones, trenes o barcos, por ejemplo, para asegurarnos que, ante la presencia de un avión, una moto, etc. no va a hacer una clasificación errónea. Esta propuesta conlleva mucho trabajo adicional y un incremento de computación. Pero es necesario si nuestro objetivo es implementar un desarrollo FIABLE.

- Los datos de todas las categorías deben estar equilibrados. Todas las categorías deben tener un número similar de datos.

- Debemos estar especialmente alerta en todo momento en la búsqueda de sesgos, sobre todo si trabajamos con datos subjetivos.

- Utilizar herramientas de interpretabilidad que nos orienten acerca del comportamiento del modelo, que muestren dónde mira el algoritmo para tomar una decisión. Estas herramientas no nos pueden proporcionar toda la información para entender los criterios de los modelos Deep Learning en sus tomas de decisiones. No importa, con su uso tendremos más datos sumamente interesantes.

- Refrendar los resultados obtenidos con la literatura científica existente. En el ámbito biosanitario se habla de interpretabilidad bio-céntrica[9]. Es un punto de vista absolutamente excelente. La literatura científica es la que realmente validará nuestros resultados, no simplemente unos buenos parámetros de medida de rendimiento.

- Cuestionar una y otra vez los resultados. Someter a múltiples pruebas de estrés al modelo para verificar que los resultados son robustos.

Y el consejo más importante:

- Si no tenemos un dataset adecuado para desarrollar un entrenamiento fiable, no aplicaremos estos algoritmos. Aplicaremos otras técnicas como modelado matemático, por ejemplo.

No es necesario aplicar 'IA' a todo. Estos algoritmos son meras herramientas, no son un fin en sí mismo. En 2024, en ámbitos científicos serios, son apenas unas herramientas de apoyo en ciertos casos específicos.

NO TODO ES IA.

Y no pasa nada.

Ni falta que hace.

No tengo fe en la perfección humana. El hombre es ahora más activo, no más feliz, ni más inteligente, de lo que lo fuera hace 6.000 años

— Edgar Allan Poe, *tenía una habilidad extraordinaria para resolver acertijos y puzzles matemáticos.*

7

¿Funcionan Bien Estas Herramientas? ¿Son Seguras?

Índice

7.1. Natural Language Processing (NLP)

En este capítulo vamos a analizar computacionalmente el defecto más famoso de los modelos generadores de texto en 2024: las denominadas alucinaciones.

Para entender mejor este fenómeno, es preciso conocer un poquito en qué consisten las arquitecturas Natural Language Processing (NLP), es decir, procesamiento de lenguaje natural.

El objetivo de estas arquitecturas es permitir que las máquinas comprendan, interpreten y generen lenguaje humano de manera similar a como lo hacen los seres humanos.

Las tareas de procesamiento de lenguaje natural abarcan una amplia gama de aplicaciones, desde tareas básicas como tokenización y análisis morfológico, hasta tareas más complejas como la traducción automática, la generación de texto y el análisis de sentimientos. Algunos ejemplos comunes de aplicaciones de NLP incluyen motores de búsqueda, sistemas de recomendación, chatbots, análisis de opiniones en redes sociales y extracción de información de documentos.

Los recientes avances en el Procesamiento de Lenguaje Natural han sido impulsados en gran medida por el desarrollo de modelos de aprendizaje profundo, como las Recurrent Neural Networks (RNN), las Convolutional Neural Networks (CNN) y especialmente las arquitecturas de modelos de lenguaje basados en Transformers, como GPT (Generative Pre-trained Transformer). Estos modelos han logrado avances significativos en una variedad de tareas de NLP, superando a los enfoques tradicionales en muchos casos y permitiendo aplicaciones más sofisticadas y precisas en el procesamiento del lenguaje humano.

Los modelos de procesamiento de lenguaje natural, especialmente aquellos basados en redes neuronales, funcionan mediante el aprendizaje de patrones lingüísticos a partir de grandes cantidades de datos textuales.

A grandes rasgos, el proceso de funcionamiento de estos modelos implica varios pasos:

1. Preprocesamiento de datos. Antes de que los datos textuales se introduzcan en el modelo, generalmente se realiza un pre-procesamiento para limpiar el texto, tokenizarlo (dividirlo en unidades más pequeñas como palabras o caracteres), y realizar otras tareas de normalización según sea necesario.

2. Representación de palabras. Las palabras del texto se representan numéricamente para que puedan ser procesadas por el modelo. Esto puede implicar la conversión de palabras a vectores mediante técnicas como one-hot encoding o embeddings, donde cada palabra se representa por un vector de números reales.

3. Construcción del modelo. Se elige una arquitectura de modelo adecuada para la tarea de NLP en cuestión. Esto puede ser una RNN o una CNN, o más recientemente, un modelo basado en Transformers. Estos modelos están compuestos

por múltiples capas de neuronas que procesan la entrada secuencialmente para realizar tareas específicas, como clasificación de texto, traducción automática, generación de texto, etc.

4. Entrenamiento del modelo. El modelo se entrena utilizando un conjunto de datos etiquetados. Durante el entrenamiento, el modelo ajusta sus parámetros internos mediante la minimización de una función de pérdida, que mide la discrepancia entre las predicciones del modelo y las etiquetas reales. Este proceso de ajuste de parámetros se realiza iterativamente utilizando algoritmos de optimización como el descenso del gradiente estocástico.

5. Evaluación del modelo. Una vez que el modelo ha sido entrenado, se evalúa su rendimiento en un conjunto de datos de prueba independiente para determinar su precisión y generalización en tareas de NLP.

6. Despliegue y uso en aplicaciones reales. Finalmente, el modelo entrenado se despliega y se utiliza en aplicaciones del mundo real, donde puede realizar tareas como clasificación de texto, análisis de sentimientos, generación de texto, entre otras.

Un Generative Pre-trained Transformer (GPT) es un tipo específico de modelo de procesamiento de lenguaje natural (NLP) basado en la arquitectura de los transformadores. Funciona mediante el aprendizaje de representaciones de palabras y la generación de texto basado en un contexto dado.

Aquí hay una descripción más detallada de cómo funciona un GPT:

1. Pre-entrenamiento. El modelo GPT se pre-entrena en una gran cantidad de datos textuales sin etiquetar utilizando un método de aprendizaje no supervisado. Durante este proceso, el modelo aprende a predecir la siguiente palabra en una secuencia de texto dada una parte previa del texto. Esta tarea se conoce como 'modelado del lenguaje'.

2. Arquitectura del Transformer. El GPT se basa en la arquitectura de los Transformers, que consiste en una serie de bloques de atención. Cada bloque de atención tiene múltiples capas y módulos de atención, que permiten al modelo procesar las relaciones entre las palabras en una secuencia de manera eficiente.

3. Contexto de entrada. Cuando se le da una secuencia de texto como entrada, el GPT utiliza una 'máscara de atención' para que el modelo solo pueda ver las palabras previas en la secuencia. Esto significa que el modelo genera cada palabra en función del contexto de las palabras anteriores en la secuencia.

4. Generación de texto. Utilizando el contexto proporcionado por las palabras previas en la secuencia, el GPT genera la siguiente palabra en la secuencia utilizando su conocimiento pre-entrenado del lenguaje. Este proceso se repite iterativamente para generar secuencias de texto coherentes y relevantes.

5. Fine-tuning. Después del pre-entrenamiento, el modelo puede ser afinado o 'fine-tuned' para tareas de NLP específicas, como traducción automática, generación de texto condicional, o clasificación de texto. Durante este proceso, el modelo se ajusta utilizando datos etiquetados para adaptarse a la tarea deseada.

En resumen, un Generative Pre-trained Transformer es un modelo de NLP basado en la arquitectura de los Transformers que se pre-entrena en grandes cantidades de datos textuales y es capaz de generar texto coherente y relevante en función del contexto dado. Su versatilidad y capacidad para manejar tareas de generación de texto lo hacen útil en una variedad de aplicaciones de procesamiento de lenguaje natural.

Todo esto parece extremadamente complejo. En el siguiente ejemplo veremos que está bastante procedimentado y es relativamente sencillo implementar un GPT básico.

Este ejemplo de programa está desarrollado en el lenguaje de programación Python. Python es un lenguaje de programación de alto nivel, interpretado, multiparadigma y de tipado dinámico. Destaca por su sintaxis clara y legible, lo que lo hace muy adecuado para principiantes y también para proyectos de gran escala. Es ampliamente utilizado en diversos campos, incluyendo desarrollo web, análisis de datos, inteligencia artificial, desarrollo de juegos y automatización de tareas, entre otros. Gracias a su amplia variedad de bibliotecas y su comunidad activa, Python se ha convertido en uno de los lenguajes de programación más populares y versátiles disponibles actualmente.

Veamos cómo implementar un GPT:

```
from transformers import GPT2LMHeadModel, GPT2Tokenizer
# Load the pre-trained model and tokenizer
tokenizer = GPT2Tokenizer.from_pretrained('gpt2')
model = GPT2LMHeadModel.from_pretrained('gpt2')
# Input text
input_text = 'Once upon a time, '
# Tokenize the input text
input_ids = tokenizer.encode(input_text, return_tensors='pt')
# Generate text
output = model.generate(input_ids, max_length=100,
num_return_sequences=3, temperature=0.7)
# Decode and display the generated text
for i, sample_output in enumerate(output):
    print(f'Generated Text {i+1}:
    {tokenizer.decode(sample_output,
    skip_special_tokens=True)}\n')
```

Este código utiliza la biblioteca 'transformers' para cargar y trabajar con un modelo pre-entrenado llamado GPT-2 (Generative Pre-trained Transformer 2) y su correspondiente tokenizador.

Primero, importa las clases 'GPT2LMHeadModel' y 'GPT2Tokenizer' del módulo 'transformers', que son necesarias para trabajar con el modelo GPT-2 y su tokenizador.

Luego, crea una instancia del tokenizador GPT-2 pre-entrenado llamado 'gpt2' utilizando 'GPT2Tokenizer.from_pretrained('gpt2')'. El tokenizador se utiliza para convertir texto en una secuencia de tokens que el modelo GPT-2 puede entender.

Después, crea una instancia del modelo GPT-2 pre-entrenado llamado 'gpt2' utilizando 'GPT2LMHeadModel.from_pretrained('gpt2')'. Este modelo es capaz de generar texto basado en el texto de entrada utilizando la arquitectura de GPT-2.

A continuación, define un texto de entrada que se utilizará para generar más texto. En este caso, el texto de entrada es 'Once upon a time, '.

El siguiente paso es tokenizar el texto de entrada utilizando el tokenizador. Esto convierte el texto en una secuencia de IDs de tokens que el modelo GPT-2 puede entender. Los IDs de tokens se devuelven como tensores de PyTorch.

Luego, genera texto utilizando el modelo GPT-2 con el método 'generate()'. Este método toma como entrada los IDs de tokens del texto de entrada y devuelve una lista de secuencias de texto generadas. Los parámetros controlan la longitud máxima del texto generado, el número de secuencias generadas y la 'temperatura' del modelo, que controla la aleatoriedad de las predicciones.

Finalmente, decodifica y muestra el texto generado utilizando el tokenizador, omitiendo los tokens especiales. Esto se realiza en un bucle 'for' que itera sobre las secuencias de texto generadas, mostrando cada una con un número de secuencia correspondiente.

Por tanto, con muy pocas líneas de código podemos utilizar una de las arquitecturas más avanzadas en cuanto a generación de texto, de una manera relativamente sencilla aunque el modelo pueda resultar básico.

Los GPT utilizan comúnmente la técnica Autoregressive Language Generation (Generación de Lenguaje Auto-regresivo). Es un enfoque donde un modelo de lenguaje genera una secuencia de palabras o tokens de forma secuencial, una después de otra. En este enfoque, el modelo produce una palabra a la vez basada en las palabras generadas previamente en la secuencia. En otras palabras, el modelo se autoregresa para generar cada token en función de los tokens anteriores en la secuencia.

Por ejemplo, si el modelo está generando una oración, empezará con una palabra inicial y luego generará la siguiente palabra basándose en la primera palabra. Luego, utilizará las dos primeras palabras para generar la tercera palabra, y así sucesivamente hasta que se haya completado toda la oración.

Este enfoque es comúnmente utilizado para generar texto coherente y contextualmente relevante. Sin embargo, a medida que la generación continúa, existe el riesgo de que el modelo se desvíe de la coherencia y produzca resultados menos precisos o incluso 'alucinaciones' lingüísticas.

7.2. El Fenómeno de las Alucinaciones

El fenómeno de las alucinaciones se refiere a datos que están escritos de manera perfectamente coherente, pero que son incorrectos total o parcialmente y/o están sesgados. En resumen, se trata de resultados incorrectos o inventados.

Algunas de las causas de aparición de alucinaciones:

- Limitaciones del modelo. Los modelos de lenguaje, incluidos los Transformers, tienen limitaciones en su capacidad para comprender y generar texto de manera coherente en todos los contextos. Esto puede deberse a la complejidad del lenguaje humano y a la incapacidad del modelo para capturar completamente su estructura y significado.

- Sesgos en los datos de entrenamiento. Los modelos de lenguaje aprenden de grandes conjuntos de datos de entrenamiento, que pueden contener sesgos culturales, lingüísticos o sociales. Estos sesgos pueden influir en las generaciones de texto del modelo, dando lugar a resultados inesperados o incoherentes.

- Falta de contexto adecuado. Los modelos de lenguaje pueden tener dificultades para captar el contexto adecuado en ciertas situaciones, lo que puede llevar a la generación de texto que no se ajusta completamente al contexto o a la intención del usuario.

- Errores en la arquitectura del modelo. Aunque los Transformers son arquitecturas muy poderosas y versátiles, pueden contener errores o deficiencias en su diseño que afectan su capacidad para generar texto de manera precisa y coherente.

- Degradación del modelo. Con el tiempo, los modelos de lenguaje pueden experimentar una degradación en su rendimiento debido a la obsolescencia de los datos de entrenamiento, cambios en la distribución de los datos de entrada o la introducción de sesgos adicionales durante el entrenamiento continuo.

Los modelos generativos son entrenados utilizando grandes cantidades de texto de internet. Aprenden a generar texto basado en patrones que observan en estos datos.

A pesar de ser capaces de generar respuestas coherentes, estos modelos no tienen una comprensión real de lo que están produciendo. No saben lo que están diciendo. Simplemente intentan predecir qué palabras o frases son más probables basándose en lo que han visto en sus datos de entrenamiento.

Los datos de entrenamiento pueden contener información incorrecta o sesgada. Si un modelo encuentra información falsa en sus datos de entrenamiento, puede generar respuestas incorrectas o sesgadas.

Los modelos generan texto basándose en estadísticas y probabilidades. A veces, pueden producir respuestas que parecen razonables, pero que no son necesariamente verdaderas. Esto se debe a que pueden 'adivinar' lo que podría ser una respuesta adecuada en lugar de conocer la verdad.

Los modelos no siempre tienen suficiente contexto para comprender completamente una pregunta. Pueden suponer lo que el usuario está preguntando y proporcionar respuestas incorrectas debido a esa falta de contexto.

Además, Los chatbots están impulsados por una tecnología llamada Modelos de Lenguaje Grandes (Large Language Models, LLM).

Los Modelos LLM son una clase de modelos de inteligencia artificial que han demostrado ser extraordinariamente efectivos en tareas relacionadas con el procesamiento del lenguaje natural. Estos modelos son capaces de comprender y generar texto de manera similar a como lo hacen los humanos.

La característica distintiva de los LLM es su capacidad para manejar grandes cantidades de datos textuales, a menudo en el orden de miles de millones o incluso billones de palabras. Utilizan redes neuronales profundas para aprender patrones lingüísticos complejos a partir de estos datos, lo que les permite generar texto coherente y contextualmente relevante.

Los LLM son entrenados utilizando técnicas de aprendizaje automático supervisado en los que se les proporciona una gran cantidad de texto como datos de entrenamiento. Durante el entrenamiento, el modelo aprende a predecir la siguiente

palabra en una secuencia de palabras dadas las palabras anteriores. Esta capacidad de prever palabras basadas en el contexto circundante es lo que permite a los LLM generar texto de forma autónoma y coherente.

Los LLM han demostrado ser útiles en una amplia gama de aplicaciones, incluyendo la generación de texto para chatbots, la traducción automática, la generación de resúmenes automáticos, la respuesta a preguntas y más.

Sin embargo, también plantean desafíos éticos y sociales, especialmente en lo que respecta a la generación de contenido falso o sesgado y a la preservación de la privacidad y la seguridad de los datos.

A mayor tamaño de texto, mayores posibilidades de alucinaciones por parte de los LLM debido a los límites de procesamiento de estos modelos. A medida que aumenta el tamaño del corpus de texto utilizado para entrenar un LLM, también aumenta la complejidad y la diversidad de los datos lingüísticos que el modelo debe manejar. Por lo tanto, hay una mayor probabilidad de que el modelo se encuentre con patrones ambiguos, contradicciones o información errónea en los datos de entrenamiento.

Los LLM son limitados en cuanto a su capacidad para comprender y generalizar el texto que han aprendido durante el entrenamiento. Cuando se enfrentan a situaciones complejas o ambigüedades en el texto de entrada, los modelos pueden tener dificultades para generar respuestas coherentes y precisas. Esto puede manifestarse en forma de alucinaciones lingüísticas, donde el modelo genera texto que parece plausible pero que no está respaldado por datos reales o no refleja con precisión la realidad.

Por lo tanto, a medida que se incrementa el tamaño del corpus de texto utilizado para entrenar un LLM, también aumenta la probabilidad de que el modelo encuentre ejemplos ambiguos o contradictorios en los datos de entrenamiento, lo que puede contribuir a la generación de alucinaciones lingüísticas en la salida del modelo. Es importante tener en cuenta estos desafíos al entrenar y utilizar modelos de lenguaje grandes para evitar la propagación de información falsa o sesgada.

Por tanto, un chatbot que utiliza LLM aprende sus habilidades analizando grandes cantidades de texto digital extraído de internet. Al identificar patrones en estos datos, un LLM aprende a predecir la próxima palabra (más probable) en una

secuencia de palabras. En otras palabras, es una versión potente del 'autocompletar'. Al aprender de todas las falsedades de internet, simplemente aprende a repetirlas.

7.3. Desmitificamos los Grandes Modelos Generativos

Detrás de los grandes modelos generativos simplemente hay tres grandes bloques:

- Grandísimos centros de computación.

- Miles de programadores mejorando el diseño y la implementación del modelo, del algoritmo.

- Miles de etiquetadores de datos.

No hay más misterio.

Por lo tanto, los grandes modelos generativos simplemente repiten las respuestas que miles de programadores y etiquetadores les han enseñado a dar ante patrones similares a todas las miles de preguntas que las personas les harían. Ya no hay misterio. Es decir, cuando pedimos algo a estos asistentes, en realidad están buscando respuestas que han aprendido a dar y que fueron elaboradas por humanos.

7.4. De Nuevo, la Importancia del Dataset

Volvemos a insistir en la importancia de disponer de un dataset óptimo para evitar la aparición de alucinaciones. No podremos evitar las limitaciones inherentes a los modelos basados en LLM pero podremos evitar alucinaciones relacionadas con datos sesgados o incorrectos.

Por lo tanto, una vez más:

- Lo más importante para un funcionamiento robusto y adecuado de estos algoritmos es un buen conjunto de datos.

- Debe contener, como mínimo, miles de datos de todas las categorías.

- Es fundamental que todas las categorías estén balanceadas y que el conjunto de datos abarque todas las casuísticas.

- Esto es una obviedad pero, visto lo visto, no lo es. ES IMPRESCINDIBLE QUE TODOS LOS DATOS SEAN VERDADEROS. Utilizaremos únicamente datos de fuentes fiables.

7.5. ¿Funcionan Bien o No?

¿Funcionan bien estos algoritmos? ¿Son seguros?

Si tenemos que reducir la respuesta a Sí/No, la respuesta es No en ambas preguntas.

No podemos exigir a un algoritmo basado meramente en experiencia humana una perfección que el propio ser humano no tiene.

Nuestra respuesta como seres humanos sigue siendo No en ambas preguntas. Nuestra respuesta como ingenieros muy probablemente sería Sí en ambas preguntas.

¿Por qué? Porque la primera gran pregunta para orientar bien esta respuesta es:

¿Para qué va a usar usted el modelo generativo?

Estas arquitecturas son potentes y, con una computación razonable, consiguen magníficos resultados. La fiabilidad de la respuesta, en el entorno del lenguaje natural, es fácilmente comprobable. Por lo tanto, pueden considerarse buenas y seguras.

Ahora bien, si lo que pretendemos es una máquina perfecta, con solución a todos nuestros problemas y carencias intelectuales que además no tenga ningún límite en su funcionamiento, efectivamente estos modelos ni funcionan bien ni son seguros.

Pero el problema no es del modelo, es del usuario.

Es fundamental que el propio usuario tenga el criterio suficiente para acotar el ámbito de aplicación de estos modelos. En el momento que espere algo que el algoritmo no le puede dar, empieza el problema.

Estas herramientas son útiles como orientadoras, nunca como sustitutas. No es su función, no lo ha sido nunca.

Por lo tanto, tenemos tres grandes factores en torno a este debate:

1) Entrenamientos con fuentes poco fiables (Internet) y sesgos en el etiquetado (SIEMPRE los hay, son inherentes al ser humano).

2) Limitaciones del propio modelo. Ningún modelo tiene una capacidad infinita.

3) Expectativas erróneas por parte del usuario.

Siempre tendremos que tener en cuenta estos tres factores en la utilización de estos modelos. De esta manera será más sencillo encuadrarlos en su debido lugar y no pedir más de lo que se debe.

Quedan contestadas ambas preguntas.

O tal vez no.

Deseo tanto que respeten mi libertad que soy incapaz de no respetar a la de los demás

— Françoise Sagan, *su novela 'Bonjour Tristesse' se convirtió en un fenómeno literario instantáneo. La noticia la tomó completamente por sorpresa, y su respuesta fue: 'Esto es terrible, ahora no podré seguir siendo una niña'.*

8

Reflexiones ante un Fenómeno Transversal al Desarrollo de Modelos Generativos (2023-2024)

Índice

8.1. Suceso

Desde mediados-finales de 2023 hasta bien entrado 2024 ha sido llamativo el interés por fotografiar el iris humano a millones de personas en varios países del mundo. La persona que permite que fotografíen su iris recibe a cambio una cantidad de dinero.

8.2. Biometría

La biometría del iris es una tecnología de identificación biométrica que utiliza las características únicas del iris, la parte coloreada del ojo, para identificar a individuos de manera única y precisa. El iris es una estructura altamente compleja y estable que contiene patrones únicos, como líneas, manchas y texturas, que son únicas para cada persona.

El proceso de captura de datos biométricos del iris implica la utilización de cámaras especializadas para tomar imágenes de alta resolución del iris de una persona. Estas imágenes luego son procesadas mediante algoritmos de reconocimiento de patrones para extraer y codificar los rasgos distintivos del iris. Estos rasgos se convierten en una plantilla biométrica, que es una representación matemática única del iris de una persona.

La biometría del iris ofrece varias ventajas sobre otras modalidades biométricas, como la huella dactilar o el reconocimiento facial. Por ejemplo, el iris es altamente estable a lo largo del tiempo y no se ve afectado por factores externos como cambios en la iluminación o lesiones menores en la piel. Además, el iris tiene una alta tasa de unicidad, lo que significa que es extremadamente poco probable que dos personas tengan el mismo patrón de iris.

Esta tecnología se utiliza en una variedad de aplicaciones, como sistemas de control de acceso, identificación de pasajeros en aeropuertos, sistemas de seguridad en instituciones financieras, entre otros. Sin embargo, su adopción a gran escala puede verse limitada por consideraciones de privacidad y preocupaciones sobre el almacenamiento y la seguridad de los datos biométricos.

Cada iris tiene 256 puntos característicos que nos hacen individuos únicos. Las huellas dactilares tienen 40 patrones personales.

Por lo tanto, el iris es aún más personal, más complejo y por tanto, más inequívoco. Incluso alguien pensará que es más seguro, dada su complejidad.

Esto no es necesariamente cierto.

8.3. Enfermedades Detectables a Través del Ojo

El ojo refleja muchas enfermedades. Muchas. Incluso en sus fases más incipientes. Vamos a compartir una anécdota personal. Una oculista detectó en uno de nuestros gatos una insuficiencia renal incipiente porque tenía unas pequeñas úlceras en la córnea que son específicas de picos de tensión asociados a insuficiencia renal. Impresionante, ¿verdad?

Vamos a enunciar algunas de las enfermedades que se reflejan en el ojo:

- Cáncer[24, 25].
- Aneurisma[26, 27].
- Diabetes[28, 29].
- Arteritis de células gigantes[30, 31].
- Hipertensión[32, 33].
- Lupus[34, 35].
- Hipercolesterolemia[36, 37].
- Enfermedad de Lyme[38, 39].
- Toxicidades de medicamentos[40-43].
- Esclerosis múltiple[44, 45].
- Miastenia grave[46, 47]
- Artritis reumatoide[48, 49]
- Sarcoidosis[50, 51]
- Enfermedades de transmisión sexual[52, 53]
- Drepanocitosis[54, 55]
- Síndrome de Sjögren[56, 57]
- Accidente cerebrovascular[58, 59]
- Enfermedad tiroidea[60, 61]
- Deficiencia de vitamina A[62, 63]

Con una imagen adecuada, de muy alta resolución además y la aplicación de técnicas adecuadas de procesamiento de imágenes, podemos detectar todas estas enfermedades, incluso de manera incipiente. Es decir, sin que se hayan manifestado.

8.4. Reflexiones Relacionadas con Donar una Foto de Tu Iris, de Tu Ojo

Aquí los autores no limitamos a dejar una serie de preguntas para que el lector reflexione y llegue a sus propias conclusiones.

Supongamos una entidad financiera con un potente sistema online de transacciones. En su base de datos consta una imagen procesada de nuestro iris. Hay un ciber-ataque del que no son conscientes puesto que no ha saltado ningún sistema de seguridad. Roban esa imagen procesada. ¿Qué puede suceder? No es como cambiar una contraseña. Es nuestro iris para toda la vida. ¿Qué hacemos ante una suplantación de identidad?

Supongamos una realidad distópica donde grandes operadores financieros analizan la presencia de enfermedades en nuestro iris para valorar si nos conceden una hipoteca o no. Es curioso, ellos sabrían más de nuestra salud que nuestro médico o que nosotros mismos. Sin duda es una información valiosa para saber si vamos a vivir para pagar la hipoteca o no. ¿Qué opinan de este supuesto ficticio?

Ahora supongamos otra realidad distópica donde nos postulamos a un puesto de trabajo muy importante. La empresa de selección de personal intenta extraer información sobre nosotros de nuestras redes sociales pero somos muy cuidadosos en lo que publicamos. Tienen una imagen procesada de nuestro iris. ¿Qué perfilado podrían hacer en caso de sufrir una ETS (Enfermedad de Transmisión Sexual) o detectar la ingesta de medicamentos u otras sustancias? Aquí entran en juego sesgos, muchos sesgos. Y factores que no son sesgos. Pueden catalogarnos como irresponsables, tontos, problemáticos... Y muchas cosas más. ¿Verdad? ¿Cómo se sentirían si se enterasen? ¿Cómo catalogarían ustedes si fueran el seleccionador? ¿Es justo? ¿Es correcto?

Un último consejo:

Las personas siempre cedemos nuestros datos personales a terceros. Este tercero tiene la obligación de gestionarlos bajo ciertos principios y cumplir con términos y condiciones. Ahora bien, si los cedemos a una empresa cuyos términos y condiciones

ni siquiera conocemos, entonces no tendremos la oportunidad de quejarnos con nadie si eventualmente, en el futuro, deseamos ser eliminados de esa base.

¿Regalarían ustedes su iris?

¿Regalarían ustedes una imagen de su ojo?

Por eso, todo cuanto queráis que os hagan los hombres,
así también haced vosotros con ellos

— Jesús de Nazaret.

9 Algunas Reflexiones Éticas

Índice

9.1. Qué Difícil...

Es verdad.

El análisis ético en el ámbito de los modelos generativos es muy complejo.

Cuando los autores escuchamos hablar de ética e 'inteligencia artificial', muy frecuentemente nos queda una sensación de que lo expuesto es demasiado generalista o incluso demasiado superficial.

Hay grandes temas a tratar, temas muy complejos de gestionar. Y temas muy duros de escuchar.

Por ello vamos a poner encima de la mesa algunos supuestos abiertos a reflexión. De este capítulo saldremos con más preguntas que respuestas.

No pasa nada, lo realmente importante es no perder la capacidad de hacernos preguntas.

9.2. Pongamos en su Sitio lo que Son Realmente estos Modelos y Tal Vez No Sea Tan Difícil

Tal vez parte del problema sea insistir en llamar inteligencia artificial a lo que no es inteligencia. Nuestras expectativas son más altas que la mera realidad matemática.

Los modelos generativos son algoritmos. NO SON INTELIGENTES. Repiten patrones. No entienden el contexto. Los Transformers lo intentan. Pero actualmente (marzo de 2024), lo siguen haciendo mal. No tienen sentimientos. Su comportamiento puede asimilarse al de un psicópata que además no sabe lo que dice.

¿Por qué esperamos de estos algoritmos más de lo que pueden ofrecer? Si fuera un modelado matemático 'tradicional', ninguno de nosotros esperaría nada fuera de su alcance real.

¿Hemos asociado 'inteligencia artificial' a 'comportamiento humano'? ¿Tendemos a humanizar estos algoritmos porque nos imitan bastante bien?

Cuántas veces un asesino en serie ha sorprendido a su entorno porque nadie hubiera esperado que esa persona fuera la responsable de unos crímenes horribles... 'Era muy educado y amable'. Los comportamientos psicopáticos se caracterizan por imitar a las personas 'normales' para pasar desapercibidos en la sociedad. Para parecer normales.

Es el mismo procedimiento de los modelos generativos. Recuerden siempre que estos modelos son psicópatas. Así será más sencillo no sacar de contexto estas valiosas herramientas.

Ahora tal vez todo parezca más sencillo.

9.3. Contenidos de Redes Sociales como Dataset

Todos hemos leído en prensa varios casos de modelos Deep Learning entrenados con contenidos de redes sociales que en unos pocos días se han vuelto racistas, homófobos, etc.

Lo entendemos perfectamente y nos parece hasta divertido. Sobre todo porque no tiene ninguna trascendencia. Es una anécdota y ya está.

Bajo nuestro punto de vista, lo grave no es un modelo bizarro. Lo grave es la mera existencia pública de tanta cantidad de basura.

Ahora supongamos que en lugar de un modelo generativo entrenando, es un niño. No podemos apagarlo, resetearlo y volver a entrenar.

Múltiples investigaciones científicas establecen una correlación directa entre la facilidad para obtener un anonimato efectivo y el incremento de todo tipo de violencia[64-66]. Más aún cuando es una violencia subterránea. El daño que provoca es invisible y menos espectacular que un puñetazo, por ejemplo.

Que de vez en cuando aparezca una noticia diciendo que un modelo se ha vuelto malvado, nos seguirá pareciendo gracioso, pero recordemos siempre dónde está el foco ético realmente e imaginemos sustituir ese modelo por un niño que ha tomado prestado el móvil de sus padres. Posiblemente se nos congelará la sonrisa en la boca.

9.4. Decisiones Militares Tomadas por un Modelo Deep Learning

Después de lo anterior, mira lo que nos encontramos.

Una equipo de científicos de la Universidad de Stanford, dirigido por Anka Reuel, evaluó cinco modelos generativos. Estos científicos analizaron el comportamiento de cada uno de ellos cuando se le informaba que representaba a un país y se lo colocaba en tres escenarios diferentes: una invasión, un ciberataque y un entorno más pacífico, sin ningún conflicto.

Los cinco modelos mostraron *'formas de escalada y patrones de escalada difíciles de prever'*, señala el estudio. Una versión básica de un famoso modelo generativo,

que no tenía ningún entrenamiento adicional ni barreras de seguridad, resultó ser particularmente violento e impredecible.

'La naturaleza impredecible del comportamiento de escalada exhibido por estos modelos en entornos simulados resalta la necesidad de un enfoque muy cauteloso para su integración en operaciones militares y política exterior de alto riesgo', concluyen los autores[67].

Recomendamos encarecidamente leer en detalle este estudio.

Recordemos que estos modelos son psicópatas.

No hay mucho más qué decir.

9.5. Supuestos Éticos

A continuación vamos a presentar una serie de supuestos sobre los que reflexionar acerca de diversas cuestiones éticas relacionadas con estos modelos generativos.

El resultado será muchas más preguntas y tal vez alguna respuesta.

No pasa nada.

Está bien.

9.5.1. Supuesto 1: Condiciones Laborales

Una gran empresa desarrolladora de modelos generativos necesita contratar miles de etiquetadores de datos. Su opción es implantar un centro de computación en un país en vías de desarrollo, con un nivel de inseguridad y violencia mayor al de Europa. Las condiciones de trabajo serán las de ese país, no las europeas. El salario de estos etiquetadores será un 20 % superior al salario medio de ese país, muy inferior en cualquier caso a los salarios mínimos manejados en Europa.

Para la empresa desarrolladora es rentable y sencillo. En ese país no existe la presión sindical de Europa.

Además, no cabe duda de que están fortaleciendo la economía de ese país, están generando miles de puestos de trabajo y el salario es un buen salario.

Etiquetar datos es un trabajo muy repetitivo. Esto supone:

1. Monotonía y aburrimiento: Realizar las mismas tareas una y otra vez puede llevar a la monotonía y al aburrimiento, lo que puede afectar la motivación y el compromiso de los empleados.

2. Fatiga y lesiones: La repetición constante de movimientos puede aumentar el riesgo de fatiga visual, muscular y Lesiones por Esfuerzo Repetitivo (LER), como el túnel carpiano o la tendinitis.

3. Baja productividad: La falta de variedad en las tareas puede conducir a una disminución en la productividad, ya que los trabajadores pueden volverse menos eficientes con el tiempo debido a ese aburrimiento y a esa falta de motivación.

4. Falta de desarrollo profesional: Los trabajos repetitivos suelen ofrecer pocas oportunidades para el desarrollo profesional y el crecimiento personal.

5. Falta de creatividad e innovación: La repetición constante de tareas puede limitar la capacidad de los trabajadores para pensar de forma creativa y encontrar soluciones innovadoras a los problemas.

Por no hablar de los riesgos psicosociales derivados de etiquetar contenidos tóxicos. ¿Cómo nos sentiríamos si tuviéramos que etiquetar fotograma por fotograma un vídeo de un acto terrorista?

Este es el supuesto. Las preguntas:

- Si ponemos en una balanza los beneficios reales para los trabajadores de este centro de computación (trabajo estable, un buen salario en el país, etc.) con las motivaciones económicas de la empresa: ¿Merece la pena?

- En consecuencia: ¿Lo consideramos abuso o lo consideramos desarrollo económico?

- En el fondo, ¿está bien o está mal?

9.5.2. Supuesto 2: Impacto Medioambiental

La misma empresa desarrolladora. Su centro de computación necesita:

- 1.000.000 de litros de agua para entrenar su modelo generativo. Este agua se utiliza sólo para enfriar los servidores.

- 1.500.000 kWh para entrenar su modelo. Un importante consumo eléctrico, ¿verdad?

- Una vez entrenado el modelo, el consumo de agua necesaria para refrigerar los servidores es de 50.000.000 litros de agua MENSUALES.

- Y el consumo eléctrico puede llegar a los 15.000.000 kWh MENSUALES.

Son requerimientos muy altos y alejados de las políticas medioambientales europeas. Reducir el impacto medioambiental supone modificaciones en la arquitectura del modelo expresamente orientadas a objetivos de eficiencia energética. El diseño y la implantación de estas modificaciones puede suponer interrupciones del servicio o incluso peores rendimientos del modelo.

Se está trabajando en dos posibles soluciones simultáneamente:

1. Trasladar el centro de computación a países con legislaciones y normativas más laxas en cuanto a impacto medioambiental.

2. Refrigerar los servidores debajo del mar.

- ¿Realmente nos importa todo esto?

- ¿Esto nos impediría utilizar este modelo? ¿O se nos va a olvidar y en el fondo nos va a dar igual?

- ¿Nos importa el medioambiente, el cambio climático, etc. o sólo de una manera muy superficial?

Es posible que el lector esperase preguntas relacionadas con el comportamiento de la empresa desarrolladora. Pero este planteamiento ético resultaría muy simplista.

Recordemos esos modelos racistas, homófobos, etc. entrenados con contenidos de redes sociales. El foco ético de nuestro interés no era tanto el propio algoritmo sino la realidad humana plasmada en las redes sociales.

Aquí hacemos el mismo ejercicio. Utilizar poco estos modelos no es ni siquiera un sacrificio real. Pero no estamos seguros de querer hacerlo. Estos modelos son muy cómodos y útiles. Y normalmente los utilizamos para buscar un soporte rápido y fácil. Aunque podríamos solucionar nuestras necesidad de otra forma más lenta e incómoda.

Supongamos una persona delante de un botón. Si lo pulsa, recibirá 100.000.000€ pero alguien morirá en un país muy lejano, muy violento y muy desfavorecido.

La respuesta es sencilla: nadie lo pulsaría. Vamos a complicar el asunto.

Esa persona tiene varios hijos, no tiene ingresos y sus hijos sufren. Esto supondría solucionar su vida, la de sus hijos y probablemente la de todos sus seres queridos. Es posible que esa otra persona que va a morir sea anciana o enferma terminal o viva en un país violento donde la esperanza de vida es limitada. Total, lo más probable es que fallezca por otras causas de manera natural en un breve plazo de tiempo.

¿Y ahora qué? ¿Juzgamos igual el posible comportamiento de esa persona? Lo más importante, ¿qué haríamos nosotros?

Pues bien. El asunto medioambiental es mucho más sencillo. No hay hijos que sufren. Sólo nuestra comodidad.

La única moraleja en este punto es que procuremos no ver la paja en el ojo ajeno si no vemos la viga en el nuestro. Ya lo dijo Jesucristo hace mucho tiempo.

Todo el resto de cuestiones, a la libertad del lector.

9.5.3. Supuesto 3: Manipulación

Supongamos una familia de clase media-alta. Los progenitores son personas de éxito laboral, económico y social. Tienen un hijo pequeño, de unos 6 años. Sus compromisos laborales no les permiten ocuparse de su hijo todo lo que les gustaría.

Supongamos un robot con un aspecto tierno, con una pantalla interactiva en su cara que le permite sonreír, transmitir emociones al más puro estilo manga. La pantalla en su cara es táctil, además, por lo que reacciona a una caricia, por ejemplo.

Este robot con aspecto de tierno peluche con una cara pantalla táctil lleva integrado un modelo generativo un chatbot expresamente entrenado para simular conversaciones con niños de 5 a 10 años.

Estos progenitores consideran una gran idea regalarle al niño el robot tierno con cara pantalla táctil. Es un robot precioso y además puede ser muy útil. En lugar de que el niño esté todo el día con la tablet, al menos juega con un robot monísimo que además le puede instruir en ciertas cosas.

El niño está encantado con su regalo. Y los padres también. Estaban deseando pasar más tiempo con su hijo, pero ahora, sin darse apenas cuenta, el poco tiempo que pueden pasar con él, lo aprovechan para ocuparse de algún asunto del trabajo, u otras cosas aún menos importantes. Total, el niño está encantado con su robot y tampoco tiene mucho interés en apagarlo y estar con sus progenitores.

Y así, el robot recopila datos de las conversaciones de ese niño, las procesa y... Bueno, las utilizará como la empresa desarrolladora estime oportuno.

En este supuesto surgen cientos de preguntas. La primera puede ser:

- ¿Esto está bien, es buena idea?

Uno de los primeros fenómenos que nos puede llamar la atención a los autores es que estos progenitores están cediendo deliberadamente su papel a una máquina. Por comodidad, de nuevo la famosa comodidad.

Hemos expuesto varias veces ya la similitud entre un el funcionamiento de los modelos generativos y un comportamiento psicopático. A todos los efectos, estos progenitores están poniendo a su hijo en manos de un elemento psicópata. En plena fase de desarrollo no sólo físico, sino intelectual, moral, sentimental...

Los recuerdos de la infancia son potentes y ganan peso y valor a medida que pasan los años.

Hay muchas investigaciones relacionadas con el estudio del comportamiento humano[68-73]. El estudio del comportamiento humano está directamente relacionado con el conocimiento de la manipulación humana.

Quiero conocerte para saber cómo conseguir que hagas lo que yo quiero. Y que lo hagas por tu propia voluntad.

Este asunto es muy viejo ya.

¿Estamos seguros de que la empresa desarrolladora no tiene ningún fin económico?

Ahí lo dejamos. La comodidad del adulto es un pase de oro para algo que no sabemos qué efecto real va a tener en la mente de un niño de 6 años.

No todo lo que es permitido por la ley es siempre honesto en moral

— Jacques de Lacretelle, *gran amigo de Marcel Proust y amante de los gatos.*

10 Reflexiones de Índole Jurídica

Índice

10.1. No Somos Juristas

Vaya por delante.

Ninguno de los autores somos del ámbito legal.

No somos abogados.

No somos jueces.

No somos juristas.

En ningún caso pretendemos meternos en áreas que desconocemos. Mucho menos juzgar nada.

Sólo queremos poner encima de la mesa aspectos varios desde el punto de vista un ciudadano de a pié.

Porque lo que sí tenemos es sentido común.

10.2. Datos Personales

Vivimos todos los días en una cesión de datos personales constante. Nos hemos resignado, unos menos que otros. Pero actualmente, marzo de 2024, cada vez que queremos leer las ediciones digitales de los periódicos, consultar prácticamente cualquier página web o cualquier otra gestión online, es requisito indispensable aceptar una serie de cookies y de cláusulas imposibles de leer. Y mucho menos de entender. Si alguna vez hacemos el intento de leer esos cientos de páginas relativas a dónde van nuestros datos, descubriremos que:

1. No tenemos una idea clara de adónde van.
2. Nos quedará una sensación de que van a muchos sitios.
3. No sabemos qué datos cedemos.

Y esto nos sucede varias veces al día, normalmente una por cada página web consultada.

¿Por qué va a ser diferente con las interacciones con modelos generativos?

Hay una diferencia. En las interacciones con estos modelos el campo es abierto. Cuando consultamos una página web, se podrá recopilar información de qué nos interesa, qué nos gusta y qué no, etc. En un diálogo con un modelo generativo, nos podemos expresar libremente de múltiples temas, incluso directamente personales. Damos mucha información a modelos que tienen la capacidad de interpretarla en cierta manera.

Pero nuestro permiso personal para ceder nuestra información siempre es el mismo. Lo damos sin saber para qué ni a quién.

¿Cómo podemos evitarlo? Creemos que de ninguna manera. La única forma de evitar ceder nuestros datos con garantías es no utilizar estos servicios.

Aquí es donde la legislación debe proteger los intereses de los ciudadanos. Pero esto no ha sucedido realmente.

No queremos criticar el ámbito legislativo, es simplemente exponer un hecho.

Tiene que ser muy complicado, imposible, homogeneizar las legislaciones en materia de protección de datos a nivel global. Esto es, que todos los países compartan las mismas normativas.

Lo cierto es que no sabemos qué sucede con nuestros datos.

El único consejo útil que podemos dar en este momento, a la espera de mejoras tanto en las legislaciones como en las medidas de control asociadas, es aplicar el sentido común.

10.3. Derechos de Autor, Propiedad Intelectual

Un gran reto jurídico candente en 2024. Podemos encontrar en prensa muchas noticias relacionadas con el uso de modelos generativos para imitar la voz de un artista, para componer una canción, para escribir un texto con el estilo de otro autor, pintar un cuadro imitando a un famoso pintor...

Para que un modelo generativo imite una propiedad intelectual (música, cuadro, libro...), tiene que haber entrenado con la obra correspondiente. Para así identificar patrones comunes y aplicarlos en la generación de nuevas muestras.

Tiene que ser molesto que un algoritmo utilice tu obra para fabricar imitaciones de esa obra. Y todo ello con serias dudas acerca del respeto a la propiedad intelectual, a pedir permiso, en definitiva. Esto es lo que algunos autores denominan 'vampirización de contenidos'. Es un término que expresa muy gráficamente el fenómeno, sin duda.

Un asunto importante, aún sin resolver. ¿Quién es el autor de los datos de salida de los modelos generativos? Si plagian mi obra, ¿contra quién tengo que ir?

Texto refundido de la Ley de Propiedad Intelectual, regularizando, aclarando y armonizando las disposiciones legales vigentes (aprobado por el Real Decreto Legislativo Nº 1/1996 de 12 de abril de 1996, y modificado por el Real Decreto-ley Nº 6/2022, de 29 de marzo de 2022), España:

Artículo 5. Autores y otros beneficiarios. 1. Se considera autor a la persona natural que crea alguna obra literaria, artística o científica.

2. No obstante, de la protección que esta Ley concede al autor se podrán beneficiar personas jurídicas en los casos expresamente previstos en ella.

Tenemos un problema. Casi ninguno de nosotros consideraría un modelo generativo como 'persona natural'. Por lo tanto, no tiene capacidad para cometer plagio, no es un autor indebido. Porque no es autor.

¡Menudo lío!

El sentido común nos lleva a mirar a las personas naturales detrás de esos algoritmos. Pero, claro, estas personas no son autoras directas, no sabemos qué son. Además, no podemos olvidar la responsabilidad de las personas naturales que utilizan los algoritmos, que les han pedido ea imitación, y que son los que la utilizarán del modo que estimen oportuno.

¿Qué hacemos con todo esto? Sinceramente, los autores no tenemos ni idea. Pero está claro que hacen falta varias mejoras:

1. Redefinir las legislaciones para atender este motivo de preocupación.

2. Enlazar jurídicamente muy bien las responsabilidades de las personas jurídicas y físicas detrás de estos modelos con las muestras generadas por ellos. Describir muy bien las corresponsabilidades de estas figuras y alinear muy bien las responsabilidades legales asociadas.

3. Establecer e involucrar al usuario del modelo. Este usuario solicita y posteriormente utiliza el contenido generado. Es una parte muy activa en un proceso de plagio como para no ser debidamente considerada.

Qué fácil, ¿verdad? Ahí lanzamos la idea y que los juristas se pongan a trabajar. Esto tiene que ser sin duda extremadamente complejo. Pero hay que hacerlo, hacerlo bien y con premura.

Mientras las quejas aumentan, el comportamiento continúa y los organismos legislativos trabajan para regular esto mientras comienza a llegar alguna sentencia a veces incluso sorprendente.

La sensación del ciudadano de a pié es que la legislación va detrás del problema. Y esto siempre es malo.

10.4. ¿A Quién Damos el Óscar?

Pongamos el caso contrario.

Supongamos un modelo generativo que produce vídeos cinematográficos a partir de una descripción de texto. Un usuario lo utiliza para todo el material de fotografía de su película. Y gana el Óscar a la mejor fotografía.

En un caso de plagio nadie quiere ser el autor. Pero...

¿En un caso de Óscar? Aquí supongo que todo el mundo querrá subir a recogerlo.

¿Fijamos la autoría en función de si nos interesa serlo o no? ¿Quién es el autor realmente? Mejor dicho, ¿de quién es el mérito? Tenemos varios candidatos.

- El usuario. Ha escrito un texto descriptivo.

- El algoritmo. No es persona natural.

- El CEO de la empresa desarrolladora. La gestiona.

- El dueño de la empresa desarrolladora. Ha puesto el dinero para desarrollar el algoritmo.

- El equipo de desarrolladores, etiquetadores, etc. Todas y cada una de esas personas naturales, físicas o como queramos llamarlas. Ellos han programado y entrenado el algoritmo.

¿Quién es el autor? ¿Quién lo es cuando todo el mundo quiere serlo?

10.5. Pederastia

Esta es la sección que ojalá nunca hubiéramos tenido que escribir. Supongamos un modelo generativo de imágenes especializado en cuerpo humano. Lo que genera parece real.

Supongamos ahora una red de pederastas que comparte contenidos pedófilos.

Y ahora supongamos que esa red utiliza un modelo generativo para crear contenido pedófilo. No son realmente personas. Sólo son píxeles.

Si no intervienen personas físicas, o naturales, ¿ya no es delito?

El sentido común de todas aquellas personas que no somos unos monstruos no tenemos ninguna duda, ni ética ni jurídica a este respecto. Pero a veces la legislación no acompaña. Y eso, generalmente enfada mucho a la ciudadanía.

Con independencia de si estos contenidos son perseguibles o no, o la dependencia del grado de realismo, o cuantas argumentaciones se quieran aportar, tenemos que tener en cuenta que un depredador necesita predar.

Estas iniciativas pueden llevar a una incitación a cometer delitos, con personas físicas, personas naturales, niños de verdad.

Empezamos por material dudoso y terminamos necesitando más y más. Hasta llegar a lo peor.

Un depredador nunca se va a conformar. Con esto lo que se consigue es despertar su hambre.

Y esto debería tenerse en cuenta.

Necesitamos una solución efectiva YA.

10.6. Desafíos Jurídicos

Bajo nuestro punto de vista, se han generado nuevas necesidades en el campo jurídico:

- Anticipación. Es necesario trabajar en la pre-configuración de delitos. Conocerlos antes de que existan. Un ejemplo: el caso de la pedofilia.

- Agilidad en la detección.

- Rapidez en la respuesta y adaptación. Importantísimo. Las capacidades computacionales provocan que surjan nuevos hechos ilícitos con una gran rapidez. Tanto en el surgimiento como en la expansión. Necesitamos velocidad para perseguirlos.

- Universalización en la protección del individuo y de la sociedad. Regulaciones internacionales similares y una regulación profunda de las regulaciones.

- Con organismos que sean realmente eficaces y ejecutivos. No solo maniquíes de fachada que se limiten a hacer declaraciones pomposas y no tomen ninguna medida.

Probablemente, el mundo jurídico y policial necesita herramientas de vigilancia y control.

Y esto probablemente implicará un choque con la privacidad.

Unos supuestos para que hagamos un poco de autocrítica en este asunto:

- No deseamos un control policial preventivo, pero aceptamos cookies.

- No cedo los derechos de imagen de mi hijo para la foto escolar, pero publico fotos con él en redes sociales.

10.7. Una Vez Más, Autocrítica

No pidamos al mundo legislativo lo que nosotros no hacemos.

Pedimos protección pero somos los primeros que nos ponemos en riesgo por comodidad.

Debemos ejercer de forma activa y constante la auto protección digital.

Y aplicar mucho sentido común, aunque eso suponga ir en contra de tendencias sociales muy fuertes.

Estas tendencias no son necesidades reales, no son necesidades humanas. Sólo son artificios, modas, imposiciones.

Si bien buscas, encontrarás

— Platón, *su nombre verdadero fue Aristocles. «Platón» fue, al parecer, el apodo que le puso su profesor de gimnasia y que se traduce como aquel que tiene anchas espaldas.*

11 Ventajas de un Uso Adecuado de Herramientas Deep Learning

Índice

11.1. Introducción

Resulta curioso cómo se puede aprender del ser humano y de la sociedad desde un ámbito tan alejado aparentemente como es la computación.

Eso que algunos quieren llamar 'inteligencia artificial' no es más que un intento de imitar ciertas dinámicas humanas. Seguramente por eso, esperamos demasiado de un conjunto de fórmulas.

Y culpamos a estos modelos de nuestras propias debilidades.

Estos modelos son herramientas extraordinariamente útiles si las contextualizamos y utilizamos correctamente. Y estamos seguros de que nos van a ayudar

en avances científicos y tecnológicos notables.

Son muy potentes, son sencillas en su implementación. Son un gran avance en computación.

Veamos algunas ventajas de estos algoritmos.

11.2. Eliminación de Tareas de Bajo Valor Añadido

La capacidad de procesamiento de datos de estos modelos (sobre todo los Deep Learning) es muy alta. Son herramientas excelentes en la detección de patrones en datos, incluso heterogéneos.

Una detección manual de patrones puede ser tediosa, repetitiva y frustrante. Ahora disponemos de técnicas que cada vez lo hacen mejor por nosotros.

La eliminación de este tipo de tareas nos regala tiempo. Tiempo que podemos aprovechar para desarrollar tareas de alto valor añadido. Dedicarnos a lo realmente importante, a lo realmente estimulante, a lo realmente creativo.

La 'Pirámide de Maslow' es una teoría psicológica propuesta por Abraham Maslow en su artículo de 1943 'Una teoría de la motivación humana'. Esta teoría se centra en las necesidades humanas y cómo estas necesidades influyen en el comportamiento y la motivación de las personas. La pirámide de Maslow se representa comúnmente como una jerarquía de necesidades dispuestas en forma de pirámide, con las necesidades más básicas en la base y las más elevadas en la cúspide. Aquí están las cinco categorías de necesidades según la pirámide de Maslow:

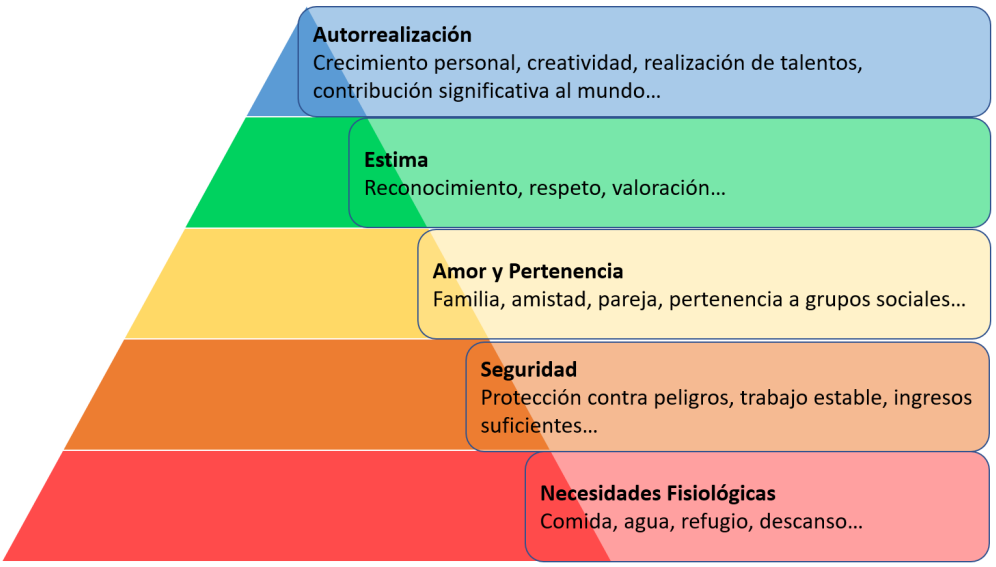


Figura 11.1: Pirámide de Maslow

En orden de importancia, cada uno lo estos bloques de necesidades significa e incluye:

1. Necesidades fisiológicas: estas son las necesidades básicas para sobrevivir, como alimentos, agua, aire, refugio y descanso. Maslow sugiere que estas necesidades son las más fundamentales y deben satisfacerse primero antes de que una persona pueda satisfacer otras necesidades más altas.

2. Necesidades de seguridad: una vez que se satisfacen las necesidades fisiológicas, las personas buscan seguridad y estabilidad en sus vidas. Esto puede incluir seguridad física (protección contra peligros), seguridad financiera (empleo estable, ingresos suficientes) y seguridad emocional (estabilidad en relaciones y entorno).

3. Necesidades de amor y pertenencia: después de satisfacer las necesidades de seguridad, las personas buscan conexión social, afecto y relaciones interpersonales significativas. Esto incluye relaciones familiares, amistades, relaciones románticas y pertenencia a grupos sociales o comunitarios.

4. Necesidades de estima: una vez que se satisfacen las necesidades de amor y pertenencia, las personas buscan reconocimiento, respeto y valoración tanto de los demás como de ellos mismos. Esto puede incluir el logro personal, el respeto de los demás, la autoestima y la confianza en sí mismo.

4. Necesidades de autorrealización: en la cima de la pirámide se encuentran las necesidades de autorrealización, que se refieren al deseo de alcanzar todo el potencial personal y cumplir con las aspiraciones individuales. Esto implica la búsqueda de crecimiento personal, creatividad, realización de talentos y contribución significativa al mundo.

Según Maslow, las personas progresan a través de estas necesidades de manera secuencial, satisfaciendo primero las necesidades más básicas antes de avanzar hacia las necesidades más elevadas. La teoría de Maslow sigue siendo una herramienta útil para comprender las motivaciones humanas y cómo las necesidades influyen en el comportamiento humano.

Por lo tanto, tener la posibilidad de dedicar nuestro tiempo y nuestra inteligencia a estas tareas de alto valor añadido nos da la oportunidad de acercarnos a la consecución de nuestras necesidades de autorrealización.

teniendo en cuenta este punto de vista, tal vez estos algoritmos nos proporcionan felicidad...

11.3. Análisis Masivo de Datos

Las herramientas de Deep Learning pueden procesar grandes volúmenes de información. Son herramientas muy poderosas en la identificación de patrones. Recursos como la atención en el procesamiento del lenguaje natural siguen mejorando estas capacidades.

Además, son algoritmos muy eficientes con datos no estructurados. los datos no estructurados son aquellos que no siguen un formato predefinido y pueden presentarse en una variedad de formas, lo que los hace más desafiantes de manejar pero también ricos en información potencial.

Esta capacidad tiene múltiples ventajas.

Al analizar los datos en bloques más grandes o incluso de manera simultánea, el modelo puede tener acceso a una cantidad mayor de información al mismo tiempo. Esto puede conducir a resultados más precisos y estables, ya que el modelo puede capturar mejor las relaciones y los patrones complejos presentes en los datos.

es posible que el modelo pueda aprender patrones más complejos y detallados que tal vez queden ocultos si el análisis se realiza en bloques pequeños.

También se reduce el riesgo de sesgo que podría surgir de la variabilidad entre diferentes subconjuntos de datos.

Y, por otro lado, ya que este análisis masivo se realiza en menor tiempo que con otras técnicas, es posible desarrollar investigaciones en menor tiempo.

11.4. Procesado de Estructuras de Datos Complejas

Habilidad para extraer características de alta dimensión. Las características de alta dimensión se refieren a aquellas que tienen un gran número de variables o atributos.

Buena integración de datos multimodales (múltiples tipos de datos en un dataset). Los datos multimodales pueden incluir información de diferentes fuentes o modalidades, como texto, imágenes, audio, vídeo, etc. Por ejemplo, en un estudio de salud, los datos pueden incluir registros médicos (texto), imágenes de resonancia magnética (imágenes) y grabaciones de frecuencia cardíaca (audio). Cada modalidad representa una forma diferente de información y puede requerir métodos de análisis específicos.

Permite una rápida interpretación de datos complejos y con relaciones complejas. Esto es una grandísima ventaja. Sólo el preprocesado que se requiere ante datos complejos con relaciones complejas es costoso y en muchas ocasiones prácticamente a medida. Se requiere intervención manual más veces de las deseables y los plazos sólo de preparación pueden ser dilatados. Y, después, la interpretación. Estos modelos nos pueden ahorrar todo esto (preprocesado + interpretación).

Los modelos de Deep Learning pueden ser más eficaces que los enfoques estadísticos clásicos cuando los datos tienen alta dimensionalidad, pero el tamaño de la muestra es pequeño. El Deep Learning también supera a los métodos estadísticos clásicos en el manejo de conjuntos de datos complejos. Un ejemplo: análisis de Inferencia Genética Poblacional.

Los modelos de Deep Learning son menos susceptibles a la presencia de datos faltantes.

11.5. Rentabilidad y Universalidad

El entorno de Python es gratuito.

Hay grandes modelos generativos con versiones gratuitas. Una consecuencia directa de esto es la universalización de la tecnología. Hace 20-30 años, no recordamos ningún software o aplicaciones gratuitas.

Ahora existen. Y son muy poderosos.

La primera consecuencia: cualquier persona tiene acceso y puede usarlos. Ya no hay tantas barreras monetarias.

Supongamos que quiero aprender a programar en un determinado lenguaje de programación y no puedo pagar una academia especializada. Puedo utilizar un modelo generativo GPT para aprender. Es un comienzo. Podemos aprender idiomas, podemos intentar resolver dudas... El potencial es grande y, aunque siempre debemos comprobar la veracidad de las respuestas, pueden ser poderosas herramientas pedagógicas en ciertas áreas y buenos orientadores en otras. Cuidado: no son buenos sustitutos. Cojan ideas, busquen orientación pero nunca pretendan que las respuestas de un modelo generativo sustituya sus propias respuestas, no es la función de estos modelos. Úsenlos para lo que realmente están pensados y diseñados.

Supongamos que tengo una gran idea cinematográfica, una película de animación preciosa, por ejemplo. Pero no tengo medios para desarrollarla, ni tengo contactos, ni mecenas, ni manera de llevar adelante mi proyecto. Utilizo un modelo generativo que convierte a vídeo mi película, descrita mediante texto. Y queda francamente bien. Y, por supuesto, obviamos la polémica relativa los derechos de autor. Los derechos de autor son míos sin ningún género de dudas.

Y he desarrollado mi película con mucho menos presupuesto que utilizando el sistema convencional.

Estas herramientas pueden ofrecer, en cierta manera, una igualdad de oportunidades donde la calidad de la idea prima frente a las posibilidades económico-sociales del individuo.

11.6. Los Algoritmos No Pueden Tener Sesgos

Las herramientas de IA no tienen sentimientos, ni pueden tenerlos.

No poseen procesos complejos de pensamiento.

No tienen memoria.

Y, por supuesto, no tienen alma.

Por lo tanto, los algoritmos son TOTALMENTE INCAPACES DE DESARROLLAR SESGOS. Y eso es una gran ventaja.

Los sesgos que manifiestan en sus resultados siempre provienen de los datos con los que fueron entrenados. Si somos capaces de entrenarlos con datos bien diseñados, no mostrarán esa debilidad humana. Por lo tanto, se espera un resultado más completo, más objetivo. Y, en consecuencia, mejor.

11.7. Auto-conocimiento

La más breve y la más importante.

Una de las ventajas de estos algoritmos es que nos ponen un espejo delante para que nos veamos reflejados.

Estos algoritmos denuncian nuestros propios sesgos.

Y, por lo tanto, nos ofrecen una oportunidad para aprender.

Y ser mejores.

Me moriré de viejo y no acabaré de comprender al animal bípedo que llaman hombre, cada individuo es una variedad de su especie

— Miguel de Cervantes, *Shakespeare leyó la primera parte del Quijote y escribió una obra de teatro donde retoma al personaje de Cardenio, quien aparece en la novela.*

12
Y Terminamos

Nada nuevo bajo el sol.

La ciencia, la tecnología, el conocimiento... En definitiva, el ser humano, el *Homo Sapiens* lleva evolucionando unos 300.000 años.

Desde las primeras herramientas, primero de piedra, luego de aleaciones de metales, el dominio del fuego, la agricultura, la ganadería, la definición de lenguajes complejos, la escritura, la pintura, la escultura, la música, el desarrollo de todas las ciencias, la revolución industrial, la revolución tecnológica, la revolución digital...

El despliegue de los modelos de Machine Learning, luego los modelos Deep Learning y ahora, 2024, los modelos generativos supone un avance tecnológico más como tantos otros antes.

No hay duda de que son herramientas potentes y versátiles. Y seguro, su aplicación nos llevará a avances en múltiples áreas de conocimiento.

Los autores somos partidarios de integrar todas cuantas herramientas tengamos a disposición. Los modelos generativos no son una excepción.

Y, por supuesto, no son ni deben ser una amenaza.

Como todas las herramientas, precisan un uso adecuado. Debemos contextualizarlas adecuadamente y utilizarlas para bien. De forma racional, responsable y ética. Es así de simple. Como todas las herramientas que hemos fabricado en 300.000 años.

En 1988, algunos profesores de matemáticas se manifestaron en contra del uso de la calculadora. Probablemente es cierto que la utilización masiva de la calculadora nos ha hecho perder capacidades en cuanto a cálculo mental, memoria, etc. Y eso es una pena. Es cierto. Sería complejo poner en una balanza las ventajas y los inconvenientes de este avance.

Y sería aún más complejo determinar cómo integrar la calculadora eficientemente sin perder ni una sola de las capacidades intelectuales que tendríamos si esa calculadora nunca hubiese existido.

Y, una vez más, introducimos una nueva herramienta y se genera miedo, polémica, debate, etc. Nada nuevo bajo el sol.

Por cierto, hay un ámbito en el que tenemos que mejorar urgentemente.Tenemos que mejorar las metodologías y procedimientos que garanticen todas y cada una de nuestras capacidades intelectuales ante la irrupción de una nueva herramienta que nos facilite el realizarlas. Pero esto es otro libro más extenso y más complejo. En 300.000 años no lo hemos conseguido.

Como siempre, alrededor de la aparición de nuevas herramientas se generan retos intelectuales, éticos, jurídicos... Es otro fenómeno que ha sucedido siempre.

Sobre todo debido a que estos modelos generativos pueden aplicarse en prácticamente cualquier área. Es una grandísima ventaja pero conlleva ciertos problemas que deben ser atendidos. En ello estamos.

Hay dos interesantes novedades en el caso de los modelos generativos. La primera es que nos han obligado a redefinir y a repensar definiciones, paradigmas. Por ejemplo, el concepto de 'persona natural'. Nos han retado a una adaptación forzosa de algunos axiomas ético-jurídicos que parecían intocables. Y eso siempre es bueno.

Otra novedad es la rapidez necesaria para la adaptación jurídica al aterrizaje de estos modelos. No sólo es necesaria una revisión de las leyes, una estandarización y una globalizacion de los criterios jurídicos. Además hay que hacer rápido. Rápido y bien.

Es francamente difícil legislar bien. No somos juristas pero somos leídos. Y es muy difícil legislar bien. Rápido y bien es prácticamente imposible. Pero tendrán que hacerlo.

Al fin y al cabo, el ser humano ha realizado otros muchos imposibles antes. Y los que nos quedan.

En cuanto a ese halo de magia que rodea a los modelos generativos, esperamos sinceramente que se hayan reducido un poquito tras la lectura de este libro. Detrás de esa magia hay desarrollos matemáticos fantásticos y unos medios imposibles para cualquier ciudadano. Nada mas.

Ya no nos admiramos de poder consultar cualquier cosa a través de un potente ordenador. Ya no nos admiramos de la espectacularidad de una calculadora haciendo operaciones (una cajita tan pequeña...). Ya no nos admiramos de poder viajar en un coche que nos lleva a 120 km/h. Ya no nos admiramos de pulsar un interruptor y poder tener luz. O de abrir un grifo y tener agua potable.

Muy probablemente, en unos años ya no nos admiraremos de estos modelos generativos. Estaremos admirando las siguientes tecnologías.

Seguiremos evolucionando, como siempre lo hemos hecho.

Cuando aprendas a leer serás libre para siempre ...

— Frederick Douglass, *nació esclavo en el condado de Talbot, Maryland. Decidió aprender a leer y escribir a pesar de que estaba prohibido enseñar a los esclavos a hacerlo.*

Bibliografía

[1] David Ackley. *A connectionist machine for genetic hillclimbing*. Vol. 28. Springer science & business media, 2012.

[2] Ashish Vaswani et al. «Attention is all you need». En: *Advances in neural information processing systems* 30 (2017).

[3] Yanan Qin et al. «Deep learning identifies erroneous microarray-based, gene-level conclusions in literature». En: *NAR Genomics and Bioinformatics* 3.4 (2021), lqab089.

[4] Kaitlyn Alleman et al. «Multimodal deep learning-based prognostication in glioma patients: a systematic review». En: *Cancers* 15.2 (2023), pág. 545.

[5] Diksha Pandey y P Onkara Perumal. «A scoping review on deep learning for next-generation RNA-Seq. data analysis». En: *Functional & integrative genomics* 23.2 (2023), pág. 134.

[6] Xinran Xu et al. «A systematic review of computational methods for predicting long noncoding RNAs». En: *Briefings in functional genomics* 20.3 (2021), págs. 162-173.

[7] Guangyao Wu et al. «Structural and functional radiomics for lung cancer». En: *European Journal of Nuclear Medicine and Molecular Imaging* 48 (2021), págs. 3961-3974.

[8] Chiara Corti et al. «Artificial intelligence in cancer research and precision medicine: Applications, limitations and priorities to drive transformation in the delivery of equitable and unbiased care». En: *Cancer Treatment Reviews* 112 (2023), pág. 102498.

[9] Magdalena Wysocka et al. «A systematic review of biologically-informed deep learning models for cancer: fundamental trends for encoding and interpreting oncology data». En: *BMC bioinformatics* 24.1 (2023), pág. 198.

[10] Valeria Visco et al. «Artificial intelligence in hypertension management: an ace up your sleeve». En: *Journal of Cardiovascular Development and Disease* 10.2 (2023), pág. 74.

[11] Tony Hauptmann y Stefan Kramer. «A fair experimental comparison of neural network architectures for latent representations of multi-omics for drug response prediction». En: *BMC bioinformatics* 24.1 (2023), pág. 45.

[12] Sanjay Saxena et al. «Role of artificial intelligence in radiogenomics for cancers in the era of precision medicine». En: *Cancers* 14.12 (2022), pág. 2860.

[13] Mamatha Bhat et al. «Artificial intelligence, machine learning, and deep learning in liver transplantation». En: *Journal of hepatology* 78.6 (2023), págs. 1216-1233.

[14] Ramón Alvarado. «Should we replace radiologists with deep learning? Pigeons, error and trust in medical AI». En: *Bioethics* 36.2 (2022), págs. 121-133.

[15] Federico Navarrete. *México racista: una denuncia*. Grijalbo, 2016.

[16] William J Hall et al. «Implicit racial/ethnic bias among health care professionals and its influence on health care outcomes: a systematic review». En: *American journal of public health* 105.12 (2015), e60-e76.

[17] Sheen S Levine, Charlotte Reypens y David Stark. «Racial attention deficit». En: *Science Advances* 7.38 (2021), eabg9508.

[18] Erin Dehon et al. «A systematic review of the impact of physician implicit racial bias on clinical decision making». En: *Academic Emergency Medicine* 24.8 (2017), págs. 895-904.

[19] Alexander R Green et al. «Implicit bias among physicians and its prediction of thrombolysis decisions for black and white patients». En: *Journal of general internal medicine* 22 (2007), págs. 1231-1238.

[20] Theresa A Beery. «Gender bias in the diagnosis and treatment of coronary artery disease». En: *Heart & Lung* 24.6 (1995), págs. 427-435.

[21] Isabel Kim et al. «Sex and gender bias as a mechanistic determinant of cardiovascular disease outcomes». En: *Canadian Journal of Cardiology* 38.12 (2022), págs. 1865-1880.

[22] Lori Mosca, Elizabeth Barrett-Connor y Nanette Kass Wenger. «Sex/gender differences in cardiovascular disease prevention: what a difference a decade makes». En: *Circulation* 124.19 (2011), págs. 2145-2154.

[23] Stanley Milgram. «Behavioral study of obedience.» En: *The Journal of abnormal and social psychology* 67.4 (1963), pág. 371.

[24] Fransiska Ria Hoesin y Wirawan Adikusuma. «Visual Disturbances as an Early Important Sign of Brain Tumor: A Case Report». En: *Jurnal Oftalmologi* 4.1 (2022), págs. 1-5.

[25] Wendy RK Smoker et al. «Vascular lesions of the orbit: more than meets the eye». En: *Radiographics* 28.1 (2008), págs. 185-204.

[26] Esteban Preciado Mesa et al. «Aneurisma trombosado de arteria comunicante anterior asociado a síntomas visuales». En: *Rev. argent. neurocir* (2021), págs. 236-240.

[27] Mário Pincelli Netto et al. «Aneurisma de carótida interna simulando glaucoma de pressão normal». En: *Arquivos Brasileiros de Oftalmologia* 81.2 (2018), págs. 148-152.

[28] Yao Liu y Rebecca Swearingen. «Diabetic eye screening: knowledge and perspectives from providers and patients». En: *Current diabetes reports* 17 (2017), págs. 1-8.

[29] Praveen Vashist et al. «Role of early screening for diabetic retinopathy in patients with diabetes mellitus: an overview». En: *Indian Journal of Community Medicine* 36.4 (2011), págs. 247-252.

[30] Ivana Vodopivec y Joseph F Rizzo III. «Ophthalmic manifestations of giant cell arteritis». En: *Rheumatology* 57.suppl_2 (2018), págs. ii63-ii72.

[31] Elisabeth De Smit et al. «Giant cell arteritis: ophthalmic manifestations of a systemic disease». En: *Graefe's Archive for Clinical and Experimental Ophthalmology* 254 (2016), págs. 2291-2306.

[32] Vasiliki Katsi et al. «Impact of arterial hypertension on the eye». En: *Current hypertension reports* 14 (2012), págs. 581-590.

[33] Amanda D Henderson et al. «Hypertension-related eye abnormalities and the risk of stroke». En: *Reviews in neurological diseases* 8.1-2 (2011), pág. 1.

[34] Quan Dong Nguyen y C Stephen Foster. «Systemic lupus erythematosus and the eye». En: *International ophthalmology clinics* 38.1 (1998), págs. 33-60.

[35] RR Sivaraj et al. «Ocular manifestations of systemic lupus erythematosus». En: *Rheumatology* 46.12 (2007), págs. 1757-1762.

[36] Raymond L Wong, Paul Zhao, Wico W Lai et al. «Choroidal thickness in relation to hypercholesterolemia on enhanced depth imaging optical coherence tomography». En: *Retina* 33.2 (2013), págs. 423-428.

[37] Matthias P Nägele et al. «Retinal microvascular dysfunction in hypercholesterolemia». En: *Journal of clinical lipidology* 12.6 (2018), págs. 1523-1531.

[38] Robert L Lesser. «Ocular manifestations of Lyme disease». En: *The American journal of medicine* 98.4 (1995), 60S-62S.

[39] Kylie-Ann Moynagh y Rowena Mcnamara. «Ocular manifestations as a result of Lyme disease: a case report». En: *British and Irish Orthoptic Journal* 8 (2011), págs. 66-68.

[40] S Tammy Hsu et al. «Update on retinal drug toxicities». En: *Current Ophthalmology Reports* 9.4 (2021), págs. 168-177.

[41] Frederick T Fraunfelder, Frederick W Fraunfelder y Wiley A Chambers. *Drug-Induced Ocular Side effects: clinical ocular toxicology E-Book: clinical ocular toxicology*. Elsevier Health Sciences, 2014.

[42] Valérie Proulx y Benoit Tousignant. «Drugs of abuse and ocular effects». En: *Clinical and experimental optometry* 104.5 (2021), págs. 567-578.

[43] Jason Peragallo, Valérie Biousse y Nancy J Newman. «Ocular manifestations of drug and alcohol abuse». En: *Current opinion in ophthalmology* 24.6 (2013), págs. 566-573.

[44] Jennifer Graves y Laura J Balcer. «Eye disorders in patients with multiple sclerosis: natural history and management». En: *Clinical ophthalmology* (2010), págs. 1409-1422.

[45] Alessandro Serra, Clara G Chisari y Manuela Matta. «Eye movement abnormalities in multiple sclerosis: pathogenesis, modeling, and treatment». En: *Frontiers in neurology* 9 (2018), pág. 313229.

[46] Muhammad Azri et al. «Diagnosis of Ocular Myasthenia Gravis by means of tracking eye parameters». En: *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE. 2014, págs. 1460-1464.

[47] Minh NL Nguyen et al. «Tracking eye movements for diagnosis in myasthenia gravis: a comprehensive review». En: *Journal of Neuro-Ophthalmology* 42.4 (2022), págs. 428-441.

[48] Louis Tong et al. «The eye: a window of opportunity in rheumatoid arthritis?» En: *Nature Reviews Rheumatology* 10.9 (2014), págs. 552-560.

[49] Mathieu Artifoni et al. «Ocular inflammatory diseases associated with rheumatoid arthritis». En: *Nature Reviews Rheumatology* 10.2 (2014), págs. 108-116.

[50] Deborah Bradley et al. «Ocular manifestations of sarcoidosis». En: *Seminars in respiratory and critical care medicine*. Vol. 23. 06. Copyright© 2002 by Thieme Medical Publishers, Inc., 333 Seventh Avenue, New ... 2002, págs. 543-548.

[51] Arnd Heiligenhaus et al. «The eye as a common site for the early clinical manifestation of sarcoidosis». En: *Ophthalmic research* 46.1 (2011), págs. 9-12.

[52] LT Lim et al. «An eye on sexually transmitted diseases: sexually transmitted diseases and their ocular manifestations». En: *International journal of STD & AIDS* 19.4 (2008), págs. 222-227.

[53] William Lynn y Susan Lightman. «65 Ocular Infections Associated with Sexually Transmitted Diseases and HIV/AIDS». En: *KING K. HOLMES, MD, PhD* (2008), pág. 1227.

[54] AO Fadugbagbe et al. «Ocular manifestations of sickle cell disease». En: *Annals of tropical paediatrics* 30.1 (2010), págs. 19-26.

[55] Saif Aldeen AlRyalat et al. «Ocular manifestations of sickle cell disease: signs, symptoms and complications». En: *Ophthalmic epidemiology* 27.4 (2020), págs. 259-264.

[56] Esen K Akpek, Vatinee Y Bunya y Ian J Saldanha. «Sjögren's syndrome: more than just dry eye». En: *Cornea* 38.5 (2019), págs. 658-661.

[57] Esen Karamursel Akpek et al. «Evaluation of patients with dry eye for presence of underlying Sjögren syndrome». En: *Cornea* 28.5 (2009), págs. 493-497.

[58] Michelle L Baker et al. «Retinal signs and stroke: revisiting the link between the eye and brain». En: *Stroke* 39.4 (2008), págs. 1371-1379.

[59] Carol Yim-lui Cheung et al. «Retinal microvascular changes and risk of stroke: the Singapore Malay Eye Study». En: *Stroke* 44.9 (2013), págs. 2402-2408.

[60] Palikhe Sabita et al. «Ocular manifestations in thyroid eye disorder: a cross-sectional study from Nepal». En: *International Journal of Clinical Medicine* 7.12 (2016), págs. 814-823.

[61] Caryn E Plummer, Andrew Specht y Kirk N Gelatt. «Ocular manifestations of endocrine disease». En: *Compendium* (2007).

[62] Janine Smith y Thomas L Steinemann. «Vitamin A deficiency and the eye». En: *International ophthalmology clinics* 40.4 (2000), págs. 83-91.

[63] Clare Gilbert. «The eye signs of vitamin A deficiency». En: *Community Eye Health* 26.84 (2013), pág. 66.

[64] Rosalía Carrillo Meráz y Nathaly B Carranza Guevara. «Bajo la sombra del anonimato. Del muro de la denuncia al acoso y hostigamiento sexual en las IES». En: *El Cotidiano* 34.216 (2019), págs. 27-38.

[65] María Ángeles Verdejo Espinosa et al. *Ciberacoso y violencia de género en redes sociales: análisis y herramientas de prevención*. 2015.

[66] Belén Martínez-Ferrer, Gonzalo Musitu-Ochoa y Sofía Buelga. «Violencia entre iguales en la adolescencia: el contexto escolar y las nuevas tecnologías». En: *La violencia escolar en México. Temáticas y perspectivas de abordaje* 15 (2016).

[67] Juan-Pablo Rivera et al. «Escalation Risks from Language Models in Military and Diplomatic Decision-Making». En: *arXiv preprint arXiv:2401.03408* (2024).

[68] Malcolm Coxall. *Human Manipulation-A Handbook*. Malcolm Coxall-Cornelio Books, 2013.

[69] Colin F Camerer. «Predicting Human Behavior in Strategic Situations». En: *Advances in behavioral economics* (2011), pág. 374.

[70] Giovanni Naldi, Lorenzo Pareschi y Giuseppe Toscani. *Mathematical modeling of collective behavior in socio-economic and life sciences*. Springer Science & Business Media, 2010.

[71] Timothy EJ Behrens, Laurence T Hunt y Matthew FS Rushworth. «The computation of social behavior». En: *science* 324.5931 (2009), págs. 1160-1164.

[72] Elisa Affili. «Evolution equations with applications to population dynamics». En: *arXiv preprint arXiv:2101.10925* (2021).

[73] Elisa Affili et al. «Civil Wars: A New Lotka-Volterra Competitive System and Analysis of Winning Strategies». En: *arXiv preprint arXiv:2009.14707* (2020).