

Construcción de

**GRANDES
MODELOS DE
LENGUAJE**

desde cero

Construcción de

GRANDES MODELOS DE LENGUAJE

desde cero

Sebastian Raschka



TÍTULOS ESPECIALES

Título original: *Build a Large Language Model (From Scratch)*

Primera edición: enero de 2026

Reservados todos los derechos. El contenido de esta obra está protegido por la Ley, que establece penas de prisión y/o multas, además de las correspondientes indemnizaciones por daños y perjuicios, para quienes reprodujeren, plagiaren, distribuyeren o comunicaren públicamente, en todo o en parte, una obra literaria, artística o científica, o su transformación, interpretación o ejecución artística fijada en cualquier tipo de soporte o comunicada a través de cualquier medio, sin la preceptiva autorización.

Authorized translation from English Language edition titled *Build a Large Language Model (From Scratch)*, published by Manning Publications.

Copyright ©2025 by Manning Publications Co.

All rights reserved.

© de la traducción: Virginia Aranda González

© EDICIONES ANAYA MULTIMEDIA (GRUPO ANAYA), 2026

Calle Valentín Beato, 21

28037 Madrid



PAPEL DE FIBRA
CERTIFICADA

ISBN: 978-84-415-5251-7

Depósito legal: M-17322-2025

Impreso en España - Printed in Spain

Agradecimientos

Escribir un libro es un proyecto de envergadura, y me gustaría expresar mi sincera gratitud a mi esposa, Liza, por su paciencia y apoyo a lo largo de este proceso. Su amor incondicional y su aliento constante han sido absolutamente esenciales.

Le estoy increíblemente agradecido a Daniel Kleine, cuyos valiosos comentarios sobre los capítulos y el código fueron excepcionales. Con su buen ojo para el detalle y sus perspicaces sugerencias, las aportaciones de Daniel han transformado indudablemente este libro en una experiencia de lectura más fluida y agradable.

También me gustaría dar las gracias al maravilloso personal de Manning Publications, incluido Michael Stephens, por las numerosas y productivas discusiones que ayudaron a dar forma a la dirección que debía tomar este libro, y a Dustin Archibald, cuyos constructivos comentarios han sido cruciales, así como su orientación en el cumplimiento de las directrices de Manning. También agradezco su flexibilidad a la hora de dar cabida a los requisitos únicos de este enfoque tan poco convencional. Debo agradecer de forma especial a Aleksandar Dragosavljević, Kari Lucke y Mike Beady por su trabajo de maquetación, y a Susan Honeywell y su equipo por perfeccionar y pulir los gráficos.

Quiero expresar mi más sincero agradecimiento a Robin Campbell y a su excelente equipo de marketing por su inestimable apoyo durante todo el proceso de redacción.

Por último, hago extensivo mi agradecimiento a los revisores: Anandaganesh Balakrishnan, Anto Aravinth, Ayush Bihani, Bassam Ismail, Benjamin Muskalla, Bruno Sonnino, Christian Prokopp, Daniel Kleine, David Curran, Dibyendu Roy Chowdhury, Gary Pass, Georg Sommer, Giovanni Alzetta, Guillermo Alcántara, Jonathan Reeves, Kunal Ghosh, Nicolas Modrzyk, Paul Silisteanu, Raul Ciotescu, Scott Ling, Sriram Macharla, Sumit Pal, Vahid Mirjalili, Vaijanath Rao y Walter Reade, por sus exhaustivas valoraciones de los borradores. Su experta atención a los detalles y sus lúcidos comentarios han sido esenciales para mejorar la calidad de este libro.

A todos los que han contribuido a este viaje, les estoy sinceramente agradecido. Vuestro apoyo, experiencia y dedicación han sido fundamentales para que este libro vea la luz. Gracias a todos.

Sobre el autor



SEBASTIAN RASCHKA lleva más de una década trabajando en machine learning e inteligencia artificial. Además de ser investigador, le apasiona la enseñanza. Es reconocido por sus libros sobre machine learning con Python y por sus contribuciones al código abierto.

Actualmente, Sebastian es ingeniero de investigación en Lightning AI, donde se especializa en la implementación y entrenamiento de grandes modelos de lenguaje (LLM). Anteriormente, fue profesor asistente en el Departamento de Estadística de la Universidad de Wisconsin-Madison, centrado principalmente en la investigación del deep learning. Puedes encontrar más información sobre él en su sitio web: <https://sebastianraschka.com>.

Sobre la imagen de cubierta

La figura de la portada de este libro, titulada *Le duchesse*, o *La duquesa*, está tomada de un libro de Louis Curmer publicado en 1841. Todas las ilustraciones contenidas en él están finamente dibujadas y coloreadas a mano.

En aquella época, era fácil identificar dónde vivía la gente y cuál era su oficio o posición en la vida únicamente por su vestimenta. Manning celebra la inventiva e iniciativa del negocio informático con portadas de libros basadas en la rica diversidad de la cultura regional de hace siglos, revivida por imágenes de colecciones como esta.

Contenidos

Agradecimientos	5
Sobre el autor.....	6
Sobre la imagen de cubierta	6
<i>Prefacio</i>	13
<i>Sobre el libro</i>	15
Quién debería leer este libro	15
Cómo está organizado este libro: una hoja de ruta.....	16
Acerca del código	17
1 <i>Comprender los grandes modelos de lenguaje</i>	19
1.1. ¿Qué es un LLM?	20
1.2. Aplicaciones de los LLM	22
1.3. Etapas de la creación y uso de LLM	23
1.4. Presentación de la arquitectura <i>Transformer</i>	25
1.5. Utilizar conjuntos de datos de gran tamaño.....	28
1.6. Un vistazo más detallado a la arquitectura GPT	30
1.7. Creando un modelo de lenguaje de gran tamaño	32
Resumen.....	33
2 <i>Trabajar con datos de texto</i>	35
2.1. Comprender las representaciones vectoriales de palabras	36
2.2. Tokenización del texto	39
2.3. Convertir tókenes en identificadores de token	42
2.4. Añadir tókenes de contexto especiales	47

2.5. Codificación por pares de símbolos	50
2.6. Muestreo de datos con una ventana deslizante	53
2.7. Crear representaciones vectoriales de tókenes.....	59
2.8. Codificación de las posiciones de las palabras.....	61
Resumen.....	65

3 *Codificar los mecanismos de atención* **67**

3.1. El problema de representar secuencias largas.....	69
3.2. Captura de dependencias de datos con mecanismos de atención ...	71
3.3. Atender a diferentes partes de la entrada con autoatención.....	72
3.3.1. Un mecanismo de autoatención sencillo sin pesos entrenables	73
3.3.2. Cálculo de los pesos de atención para todos los tókenes de entrada.....	78
3.4. Implementación de la autoatención con pesos entrenables	81
3.4.1. Calcular paso a paso los pesos de atención	82
3.4.2. Implementación de una clase Python compacta de autoatención	87
3.5. Ocultar palabras futuras con atención causal.....	91
3.5.1. Aplicación de una máscara de atención causal	92
3.5.2. Enmascaramiento de pesos de atención adicionales con <i>dropout</i>	95
3.5.3. Implementación de una clase de atención causal compacta ...	97
3.6. Ampliar la atención de una sola cabeza a múltiples cabezas	99
3.6.1. Apilar múltiples capas de atención de una sola cabeza	99
3.6.2. Implementación de la <i>multi-head attention</i> con divisiones de pesos.....	103
Resumen.....	108

4 *Implementar un modelo GPT desde cero para generar texto* **109**

4.1. Codificar una arquitectura LLM.....	110
4.2. Normalización de activaciones con normalización de capas.....	116
4.3. Implementación de una red <i>feedforward</i> con activaciones GELU...	122

4.4. Incorporar conexiones de atajo.....	126
4.5. Conexión entre capas de atención y capas lineales en un bloque <i>Transformer</i>	130
4.6. Codificación del modelo GPT	134
4.7. Generando texto.....	139
Resumen.....	144

5 Realizar preentrenamiento con datos no etiquetados 145

5.1. Evaluar modelos de texto generativos.....	146
5.1.1. Utilizar GPT para generar texto.....	147
5.1.2. Cálculo de la pérdida de generación de texto	149
5.1.3. Cálculo de las pérdidas de los conjuntos de entrenamiento y validación	157
5.2. Entrenar un LLM.....	163
5.3. Estrategias de decodificación para controlar la aleatoriedad.....	168
5.3.1. Escalado de temperatura	169
5.3.2. Muestreo <i>top-k</i>	172
5.3.3. Modificación de la función de generación de texto	174
5.4. Cargar y guardar los pesos del modelo en PyTorch	176
5.5. Carga de los pesos preentrenados de OpenAI	177
Resumen.....	184

6 Ajuste fino por clasificación 185

6.1. Diferentes categorías de ajuste fino.....	186
6.2. Preparación del conjunto de datos	188
6.3. Creación de los cargadores de datos	191
6.4. Inicializar un modelo con pesos preentrenados	197
6.5. Añadir una cabeza de clasificación.....	199
6.6. Cálculo de la precisión y pérdida de clasificación.....	206
6.7. Ajuste fino del modelo con datos supervisados	211
6.8. Utilización del LLM como clasificador de spam	216
Resumen.....	218

7	<i>Afinamiento para seguir instrucciones</i>	219
7.1.	Introducción al ajuste fino por instrucciones.....	220
7.2.	Preparación de un conjunto de datos para el ajuste fino supervisado por instrucciones.....	222
7.3.	Organización de los datos en lotes de entrenamiento.....	226
7.4.	Creación de cargadores de datos para un conjunto de datos de instrucciones.....	238
7.5.	Carga de un LLM preentrenado	241
7.6.	Ajuste fino del LLM a los datos de las instrucciones.....	244
7.7.	Extracción y almacenamiento de respuestas	248
7.8.	Evaluación del LLM afinado.....	253
7.9.	Conclusiones.....	262
7.9.1.	¿Qué viene a continuación?.....	262
7.9.2.	Mantenerse al día en un campo en rápida evolución.....	263
7.9.3.	Últimas palabras.....	263
	Resumen.....	264

Apéndice A. Introducción a PyTorch **265**

A.1.	¿Qué es PyTorch?	265
A.1.1.	Los tres componentes esenciales de PyTorch.....	266
A.1.2.	Definición de deep learning.....	266
A.1.3.	Instalación de PyTorch.....	268
A.2.	Comprender los tensores.....	272
A.2.1.	Escalares, vectores, matrices y tensores.....	272
A.2.2.	Tipos de datos de tensor	273
A.2.3.	Operaciones habituales con tensores en PyTorch.....	274
A.3.	Visualización de modelos como gráficos computacionales.....	275
A.4.	La diferenciación automática simplificada.....	277
A.5.	Implementación de redes neuronales multicapa	279
A.6.	Configuración de cargadores de datos eficientes	284
A.7.	Un bucle de entrenamiento habitual	288
A.8.	Guardar y cargar modelos	292

A.9. Optimización del rendimiento del entrenamiento con GPU	293
A.9.1. Cálculos PyTorch en dispositivos GPU	293
A.9.2. Entrenamiento con una sola GPU.....	294
A.9.3 Entrenar con varias GPU.....	296
Resumen.....	302
 <i>Apéndice B. Referencias y lecturas adicionales</i>	 303
 <i>Apéndice C. Soluciones a los ejercicios</i>	 315
 <i>Apéndice D. Incorporación de funcionalidades adicionales al bucle de entrenamiento</i>	 327
D.1. Calentamiento de la tasa de aprendizaje.....	328
D.2. Atenuación cosenoidal	330
D.3. Recorte de gradientes.....	331
D.4. La función de entrenamiento modificada	333
 <i>Apéndice E. Ajuste fino eficiente en parámetros con LoRA</i>	 337
E.1. Introducción a LoRA	337
E.2. Preparación del conjunto de datos	339
E.3. Inicialización del modelo	341
E.4. Ajuste fino eficiente en parámetros con LoRA	343
 <i>Índice alfabético</i>	 351

Prefacio

Siempre me han fascinado los modelos lingüísticos. Hace más de una década, mi viaje por la IA comenzó con una clase sobre clasificación de patrones estadísticos, que me llevó a mi primer proyecto independiente: desarrollar un modelo y una aplicación web para detectar el estado de ánimo de una canción basándome en su letra.

En 2022, con el lanzamiento de ChatGPT, los grandes modelos de lenguaje o LLM (*Large Language Models*) han tomado el mundo al asalto y han revolucionado la forma de trabajar de muchos de nosotros. Estos modelos son increíblemente versátiles y ayudan en tareas como revisión gramatical, redacción de correos electrónicos, resumen de documentos extensos y mucho más. Esto se debe a su capacidad para analizar y generar texto similar al de las personas, algo de enorme importancia en varios campos, desde la atención al cliente a la creación de contenidos, e incluso en dominios más técnicos, como la codificación y el análisis de datos.

Como su nombre indica, los LLM son grandes o de gran tamaño, y abarcan entre millones y miles de millones de parámetros (a modo de comparación, utilizando métodos de machine learning o estadísticos más tradicionales, el conjunto de datos flor Iris puede clasificarse con una precisión superior al 90 % utilizando un modelo reducido con tan solo dos parámetros). Sin embargo, a pesar del gran tamaño de los LLM en comparación con otros métodos más tradicionales, no tienen por qué ser algo incomprensible.

En este libro aprenderás a construir un LLM paso a paso. Cuando llegues al final, tendrás una profunda comprensión de cómo funcionan estos modelos (como los utilizados en ChatGPT) a un nivel básico. Creo que desarrollar confianza con los conceptos fundamentales y el código subyacente es crucial para el éxito. No solamente ayuda a corregir errores y mejorar el rendimiento, sino que también permite experimentar con nuevas ideas.

Hace varios años, cuando empecé a trabajar con LLM, tuve que aprender a implementarlos por las malas, rebuscando entre muchos artículos de investigación y repositorios de código incompletos para poder comprenderlos de una forma general. Con este libro, donde desarrollo y comparto un tutorial de implementación paso a paso que detalla los principales componentes y fases de desarrollo de un LLM, espero lograr que estos modelos de lenguaje sean más accesibles.

Creo firmemente que la mejor manera de entender los grandes modelos de lenguaje es programar uno desde cero. ¡Verás lo divertido que puede llegar a ser!

¡Feliz lectura y programación!

Sobre el libro

Este libro se ha escrito para ayudarte a comprender y crear tus propios grandes modelos de lenguaje (LLM) de tipo GPT desde cero. El libro comienza centrándose en los fundamentos del trabajo con datos de texto y la codificación de mecanismos de atención y, a continuación, te guía a través de la implementación de un modelo GPT completo desde el principio. Posteriormente aborda el mecanismo de preentrenamiento, así como el ajuste fino para tareas específicas, como la clasificación de textos y el seguimiento de instrucciones. Al final tendrás una comprensión profunda de cómo funcionan los LLM y dispondrás de las habilidades necesarias para construir tus propios modelos. Aunque dichos modelos sean de menor escala en comparación con los grandes modelos fundacionales, utilizan los mismos conceptos y sirven como potentes herramientas educativas para comprender los mecanismos y técnicas básicos utilizados en la construcción de los LLM más avanzados.

Quién debería leer este libro

Este libro está dirigido a entusiastas del machine learning, ingenieros, investigadores, estudiantes y profesionales que deseen comprender en profundidad cómo funcionan los LLM y aprender a construir sus propios modelos desde cero. Tanto los principiantes como los desarrolladores experimentados podrán hacer uso de sus habilidades y conocimientos para comprender los conceptos y técnicas utilizados en la creación de LLM.

Lo que distingue a este libro es su exhaustiva cobertura de todo el proceso de creación de LLM, desde el trabajo con conjuntos de datos hasta la implementación de la arquitectura del modelo, su preentrenamiento con datos no etiquetados y su ajuste fino para tareas específicas. En el momento de escribir estas líneas, no existe ningún otro recurso que ofrezca un enfoque tan completo y práctico para construir LLM desde cero.

Para entender los ejemplos de código de este libro, debes disponer de sólidos conocimientos de programación en Python. Aunque puede venirte muy bien tener cierta familiaridad con el aprendizaje automático o machine learning (ML), el aprendizaje profundo o deep learning (DL) y la inteligencia artificial, no se requiere una amplia experiencia en estas áreas. Los LLM son un subconjunto único de la IA por lo que, incluso aunque seas relativamente nuevo en el campo, podrás entenderlo todo sin problemas.

Si tienes experiencia con redes neuronales profundas, puede que ciertos conceptos te resulten más familiares, porque los LLM se basan en estas arquitecturas. Sin embargo, el dominio de PyTorch no es un requisito previo. El apéndice A ofrece una breve introducción a PyTorch, con la que aprenderás las habilidades necesarias para comprender los ejemplos de código del libro.

A la hora de explorar el funcionamiento interno de los LLM, puede ser útil tener conocimientos de matemáticas a nivel de bachillerato, en particular del trabajo con vectores y matrices, aunque no es necesario que dichos conocimientos sean avanzados para comprender los conceptos e ideas clave de este libro.

El requisito previo más importante es una sólida base de programación en Python. Con este conocimiento, estarás bien preparado para explorar el fascinante mundo de los LLM y comprender los conceptos y ejemplos de código presentados en este libro.

Cómo está organizado este libro: una hoja de ruta

Este libro está diseñado para ser leído secuencialmente, pues cada capítulo se basa en los conceptos y técnicas introducidos en los anteriores. El libro está dividido en siete capítulos, que cubren los aspectos esenciales de los LLM y su implementación.

El capítulo 1 ofrece una completa introducción sobre los conceptos fundamentales de los LLM. Explora la arquitectura *Transformer*, que constituye la base de modelos LLM como los utilizados en la plataforma ChatGPT.

El capítulo 2 presenta un plan para construir un LLM desde cero. Trata todo el proceso de preparación del texto para el entrenamiento del LLM, incluyendo la división del texto en tokens de palabras y subpalabras, el uso de codificación de pares de símbolos para tokenización avanzada, el muestreo de ejemplos de entrenamiento con un enfoque de ventana deslizante, y la conversión de tokens en vectores que alimentan al LLM.

El capítulo 3 aborda los mecanismos de atención utilizados en los LLM. Introduce una estructura básica de autoatención y va avanzando hacia un mecanismo de autoatención mejorado. También explica la implementación de un módulo de atención causal, que permite a los LLM generar un token cada vez, enmascarando pesos de atención seleccionados aleatoriamente mediante *dropout* para reducir el sobreajuste, y apilando varios módulos de atención causal en un solo módulo de *multihead attention*.

El capítulo 4 se centra en la codificación de un LLM tipo GPT, que pueda entrenarse para generar texto similar al de las personas. Abarca técnicas como la normalización de las activaciones de las capas para estabilizar el entrenamiento de las redes neuronales, la incorporación de conexiones de atajo en redes neuronales profundas para entrenar

los modelos de forma más eficaz, la implementación de bloques *Transformer* para crear modelos GPT de varios tamaños y el cálculo del número de parámetros y los requisitos de almacenamiento de los modelos GPT.

El capítulo 5 aborda el proceso de preentrenamiento de los LLM. Abarca el cálculo de las pérdidas de los conjuntos de entrenamiento y validación para evaluar la calidad del texto generado por el LLM, la implementación de una función de entrenamiento y el preentrenamiento del LLM, el almacenamiento y la carga de los pesos del modelo para continuar entrenando un LLM, y la carga de pesos preentrenados de OpenAI.

El capítulo 6 presenta diferentes enfoques de afinamiento del LLM. Explica la preparación de un conjunto de datos para la clasificación de texto, la modificación de un LLM preentrenado para su ajuste fino, el afinamiento de un LLM para identificar mensajes de *spam* y la evaluación de la precisión de un clasificador LLM ya afinado.

El capítulo 7 explora el proceso de afinamiento de los LLM para seguir instrucciones. Incluye la preparación de un conjunto de datos, al que después se aplicará ajuste fino supervisado por instrucciones, la organización de los datos de instrucciones en lotes de entrenamiento, la carga de un LLM preentrenado y su afinamiento para seguir instrucciones humanas, la extracción de las respuestas a instrucciones generadas por el LLM para su valoración y la evaluación de un LLM afinado por instrucciones.

Acerca del código

Todos los ejemplos de código fuente de este libro están disponibles para su descarga en la página web de Anaya Multimedia en <https://anayamultimedia.es>, en la opción Selecciona complemento que encontrará en la ficha correspondiente a este libro. También puede descargarlos de la página web del libro original en <https://www.manning.com/books/build-a-large-language-model-from-scratch>, así como en formato Jupyter Notebook en GitHub en <https://github.com/rasbt/LLMs-from-scratch>. No te preocupes si te quedas atascado: en el apéndice C encontrarás las soluciones a todos los ejercicios.

Este libro contiene muchos ejemplos de código fuente, tanto en listados numerados como incluidos dentro del texto de cada capítulo. En ambos casos, el código fuente está formateado en una fuente monoespacial como esta para distinguirlo del texto normal.

Uno de los objetivos clave de este libro es la accesibilidad. Por ello, los ejemplos de código se han diseñado cuidadosamente para que se ejecuten de forma eficiente en un ordenador portátil normal, sin necesidad de ningún hardware especial. Si tienes acceso a una GPU, algunas secciones ofrecen consejos útiles sobre cómo ampliar los conjuntos de datos y los modelos para aprovechar esa potencia adicional.

A lo largo del libro, utilizaremos PyTorch como tensor de referencia y una biblioteca de deep learning para implementar LLM desde cero. Si PyTorch es nuevo para ti, te recomiendo que empieces con el apéndice A, que proporciona una detallada introducción, con recomendaciones de configuración.

