

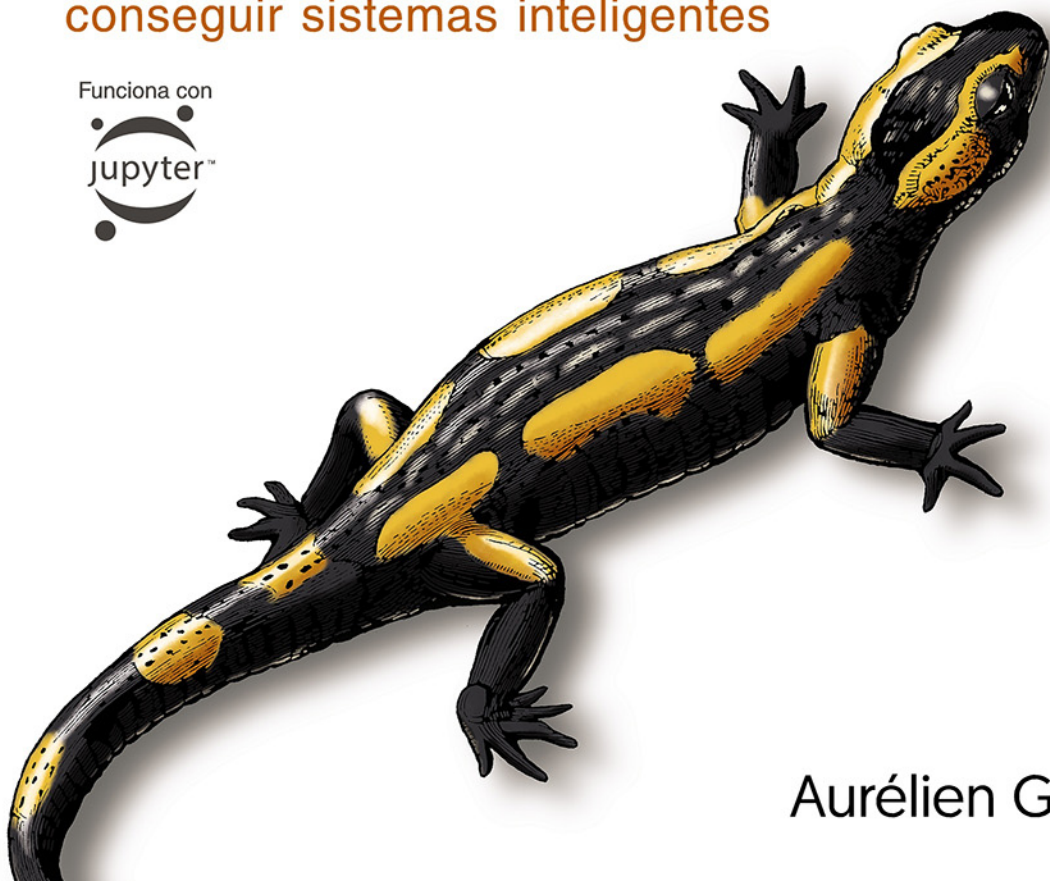
O'REILLY® ANAYA
MULTIMEDIA

TERCERA
EDICIÓN

Aprende Machine Learning con Scikit-Learn, Keras y TensorFlow

Conceptos, herramientas y técnicas para
conseguir sistemas inteligentes

Funciona con



Aurélien Géron

TERCERA
EDICIÓN

Aprende Machine Learning con Scikit-Learn, Keras y TensorFlow

Conceptos, herramientas y técnicas
para conseguir sistemas inteligentes

Aurélien Géron



ANAYA
MULTIMEDIA

TÍTULO ESPECIAL

Título de la obra original:

Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow. Third Edition

Traductor:

Beatriz Pineda González

Responsable editorial:

Víctor Manuel Ruiz Calderón

Adaptación de cubierta:

Celia Antón Santos

Primera edición digital: 2023

Todos los nombres propios de programas, sistemas operativos, equipos hardware, etc. que aparecen en este libro son marcas registradas de sus respectivas compañías u organizaciones.

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 published by O'Reilly Media, Inc.
Copyright © 2023 Aurélien Géron. All rights reserved.

Edición española:

© EDICIONES ANAYA MULTIMEDIA (GRUPO ANAYA, S.A.), 2023

Valentín Beato, 21. 28037 Madrid

ISBN: 978-84-415-4828-2

Edición electrónica sobre la 3.^a edición impresa

Agradecimientos

Nunca, ni en mis mejores sueños, imaginé que la primera y la segunda edición de este libro tendrían un público tan grande. Recibí muchos mensajes de lectores, muchos hacían preguntas, algunos indicaban con amabilidad la presencia de erratas y la mayoría enviaban palabras de ánimo. No puedo expresar lo agradecido que estoy a todos esos lectores por todo el apoyo. ¡Muchísimas gracias a todos! Por favor, no dudes en informar de cualquier problema en GitHub (<https://homl.info/issues3>) si encuentras errores en los ejemplos de código (o en hacer preguntas sin más) o de señalar erratas (<https://homl.info/errata3>) si encuentras errores en el texto. Algunos lectores compartieron su experiencia sobre cómo este libro les ayudó a conseguir su primer trabajo o a resolver un problema concreto en el que estaban trabajando. Ese tipo de comentarios son muy motivadores para mí. Si este libro te resulta útil, me encantaría que compartieses tu historia conmigo, ya sea en privado (por ejemplo, a través de LinkedIn (<https://linkedin.com/in/aurelien-geron>)) o en público (por ejemplo, con un tuit en @aureliengeron o con una reseña en Amazon (<https://homl.info/amazon3>)).

Muchísimas gracias también a todas las personas maravillosas que me ofrecieron su tiempo y su experiencia para revisar esta tercera edición, corregir errores y hacer innumerables sugerencias. Esta edición es mucho mejor gracias a ellos: Olzhas Akpambetov, George Bonner, François Chollet, Siddha Ganju, Sam Goodman, Matt Harrison, Sasha Sobran, Lewis Tunstall, Leandro von Werra y mi querido hermano Sylvain. ¡Sois increíbles!

También estoy muy agradecido a las muchas personas que me han apoyado en este camino respondiendo a mis preguntas, sugiriendo mejoras y contribuyendo al código en GitHub: en especial, Yannick Assogba, Ian Beaugregard, Ulf Bissbort, Rick Chao, Peretz Cohen, Kyle Gallatin, Hannes Hapke, Victor Khaustov, Soonson Kwon, Eric Lebigot, Jason Mayes, Laurence Moroney, Sara Robinson, Joaquín Ruales y Yuefeng Zhou.

Este libro no existiría sin el fantástico personal de O'Reilly, en especial Nicole Taché, que me ofreció unas opiniones muy perspicaces y siempre fue amable, alentadora y servicial: no podría pedir una editora mejor. Muchísimas gracias también a Michele Cronin, que me animó en los capítulos finales y consiguió hacerme cruzar la línea de meta. Gracias a todo el equipo de producción, sobre todo a Elizabeth Kelly y Kristen Brown. Gracias también a Kim Cofer por la corrección exhaustiva y a Johnny O'Toole, que gestionó la relación con Amazon y respondió a muchas de mis preguntas. Gracias a Kate Dullea por mejorar muchísimo mis ilustraciones. Gracias a Marie Beaugureau, Ben Lorica, Mike Loukides y Laurel Ruma por creer en este proyecto y ayudarme a definir su alcance. Gracias a Matt Hacker y a todo el equipo de Atlas por responder a mis preguntas técnicas acerca del formato, AsciiDoc, MathML y LaTeX, y gracias también a Nick Adams, Rebecca Demarest, Rachel Head, Judith McConville, Helen Monroe, Karen Montgomery, Rachel Roumeliotis y a todas las personas de O'Reilly que han contribuido a la publicación de este libro.

Nunca olvidaré a las personas maravillosas que me ayudaron con la primera y la segunda edición de este libro: amigos, colegas, expertos, incluidos muchos miembros del equipo de TensorFlow. La lista es larga: Olzhas Akpambetov, Karmel Allison, Martin Andrews, David

Andrzejewski, Paige Bailey, Lukas Biewald, Eugene Brevdo, William Chargin, François Chollet, Clément Courbet, Robert Crowe, Mark Daoust, Daniel "Wolff" Dobson, Julien Dubois, Mathias Kende, Daniel Kitachewsky, Nick Felt, Bruce Fontaine, Justin Francis, Goldie Gadde, Irene Giannoumis, Ingrid von Glehn, Vincent Guilbeau, Sandeep Gupta, Priya Gupta, Kevin Haas, Eddy Hung, Konstantinos Katsiapis, Viacheslav Kovalevskyi, Jon Krohn, Allen Lavoie, Karim Matrah, Grégoire Mesnil, Clemens Mewald, Dan Moldovan, Dominic Monn, Sean Morgan, Tom O'Malley, James Pack, Alexander Pak, Haesun Park, Alexandre Passos, Ankur Patel, Josh Patterson, André Susano Pinto, Anthony Platanios, Anosh Raj, Oscar Ramirez, Anna Revinskaya, Saurabh Saxena, Salim Sémaoune, Ryan Sepassi, Vitor Sessak, Jiri Simsa, Iain Smears, Xiaodan Song, Christina Sorokin, Michel Tessier, Wiktor Tomczak, Dustin Tran, Todd Wang, Pete Warden, Rich Washington, Martin Wicke, Edd Wilder-James, Sam Witteveen, Jason Zaman, Yuefeng Zhou y mi hermano Sylvain.

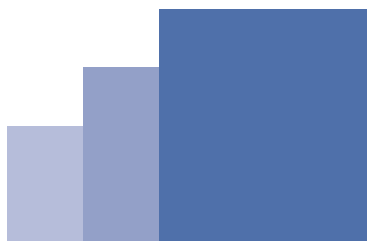
Por último, pero no por eso menos importante, estoy infinitamente agradecido a mi querida esposa, Emmanuelle, y a nuestros tres maravillosos hijos, Alexandre, Rémi y Gabrielle, por animarme a trabajar duro en este libro. Su curiosidad insaciable no tiene precio: explicar algunos de los conceptos más difíciles de este libro a mi mujer y mis hijos me ayudó a aclarar mis pensamientos y, directamente, mejoró muchas partes de esta obra. ¡Además, me traen galletas y café todo el rato! ¿Qué más se puede pedir?

Sobre el autor

Aurélien Géron es asesor y profesor de *machine learning*. Antiguo *googler*, dirigió el equipo de clasificación de imágenes de YouTube entre 2013 y 2016. Ha sido fundador y CTO de varias empresas diferentes: Wifirst, proveedor de servicios de Internet inalámbrico líder en Francia; Polyconseil, una asesoría centrada en las telecomunicaciones, los medios y la estrategia; y Kiwisoft, una asesoría centrada en el *machine learning* y la privacidad de datos.

Antes de todo eso, Aurélien trabajó como ingeniero en diversos campos: finanzas (JP Morgan y Société Générale), defensa (Departamento de Defensa de Canadá) y sanidad (transfusiones de sangre). También publicó algunos libros técnicos (sobre C++, WiFi y arquitecturas de Internet) y dio clases de informática en una escuela de ingeniería francesa.

Curiosidades: enseñó a sus tres hijos a contar en binario con los dedos (hasta 1.023), estudió microbiología y genética evolutiva antes de pasarse a la ingeniería de software y su paracaídas no se abrió la segunda vez que saltó.



Índice de contenidos

Agradecimientos	5
Sobre el autor	6

Introducción **21**

El tsunami del <i>machine learning</i>	21
<i>Machine learning</i> en tus proyectos	21
Objetivo y enfoque	22
Ejemplos de código	23
Prerrequisitos	23
Hoja de ruta	24
Cambios entre la primera y la segunda edición	25
Cambios entre la segunda y la tercera edición	25
Otros recursos	26
Convenciones	27
Sobre la imagen de cubierta	28

Parte I. Fundamentos del machine learning **29**

Capítulo 1. El paisaje del machine learning **31**

¿Qué es el <i>machine learning</i> ?	32
¿Por qué utilizar <i>machine learning</i> ?	33
Ejemplos de aplicaciones	35
Tipos de sistemas de <i>machine learning</i>	37
Supervisión del entrenamiento	37
Aprendizaje por lotes frente a aprendizaje <i>online</i>	44

Aprendizaje basado en instancias frente a aprendizaje basado en modelos.....	48
Principales retos del <i>machine learning</i>	53
Cantidad insuficiente de datos de entrenamiento.....	54
Datos de entrenamiento no representativos.....	55
Datos de mala calidad	56
Características irrelevantes.....	57
Sobreajustar los datos de entrenamiento	57
Subajustar los datos de entrenamiento.....	59
Retroceder.....	60
Probar y validar	60
Ajuste de hiperparámetros y selección del modelo.....	61
Discrepancia de datos	62
Ejercicios	64

Capítulo 2. Proyecto de *machine learning* de principio a fin 65

Trabajar con datos reales.....	65
Tener una visión amplia.....	67
Enmarcar el problema	67
Seleccionar una medida del rendimiento	69
Comprobar las suposiciones	71
Obtener los datos	72
Ejecutar los ejemplos de código usando Google Colab.....	72
Guardar los cambios en tu código y tus datos	74
El poder y el peligro de la interactividad.....	75
Código en el libro frente a código en los cuadernos	75
Descargar los datos	76
Echar un vistazo rápido a la estructura de los datos	77
Crear un conjunto de prueba	81
Explorar y visualizar los datos para tener un mayor entendimiento	86
Visualizar datos geográficos.....	86
Buscar correlaciones.....	88
Experimentar con combinaciones de atributos	91
Preparar los datos para algoritmos de <i>machine learning</i>	92
Limpiar datos	93
Manejar atributos de texto y categóricos.....	96
Escalado de características y transformación	99
Transformadores personalizados.....	103
Pipelines de transformación	106

Seleccionar un modelo y entrenarlo	111
Entrenar y evaluar con el modelo de entrenamiento	111
Una evaluación mejor utilizando la validación cruzada	113
Perfeccionar el modelo	114
Búsqueda exhaustiva	115
Búsqueda aleatorizada	116
Métodos de ensamblaje	118
Analizar los mejores modelos y sus errores	118
Evaluar el sistema en el conjunto de prueba	119
Lanzar, monitorizar y mantener el sistema	120
¡Pruéballo!	123
Ejercicios	124

Capítulo 3. Clasificación 125

MNIST	125
Entrenar un clasificador binario	127
Medidas de rendimiento	128
Medir la precisión con validación cruzada	129
Matrices de confusión	130
Precisión y sensibilidad	131
Compensación precisión/sensibilidad	133
La curva ROC	136
Clasificación multiclase	140
Análisis de errores	143
Clasificación multietiqueta	146
Clasificación multisalida	148
Ejercicios	149

Capítulo 4. Entrenar modelos 151

Regresión lineal	152
La ecuación normal	154
Complejidad computacional	157
Descenso de gradiente	158
Descenso de gradiente por lotes	161
Descenso de gradiente estocástico	164
Descenso de gradiente por minilotes	167
Regresión polinomial	168
Curvas de aprendizaje	170

Modelos lineales regularizados	174
Regresión de arista	174
Regresión Lasso	176
Regresión de red elástica	179
Detención temprana	180
Regresión logística	181
Calcular probabilidades	182
Entrenamiento y función de pérdida	183
Límites de decisión	184
Regresión <i>softmax</i>	187
Ejercicios	191

Capítulo 5. Máquinas de vectores soporte 193

Clasificación SVM lineal	193
Clasificación de margen blando	194
Clasificación SVM no lineal	196
<i>Kernel</i> polinomial	197
Características de similitud	198
<i>Kernel</i> de función de base radial gaussiana	199
Clases SVM y complejidad computacional	201
Regresión SVM	202
Entre bambalinas de los clasificadores SVM lineales	204
El problema dual	206
SVM kernelizadas	207
Ejercicios	210

Capítulo 6. Árboles de decisión 211

Entrenar y visualizar árboles de decisión	211
Hacer predicciones	212
Estimación de probabilidades de clase	215
El algoritmo de entrenamiento CART	215
Complejidad computacional	216
¿Impureza de Gini o entropía?	216
Hiperparámetros de regularización	217
Regresión	219
Sensibilidad a la orientación del eje	221
Los árboles de decisión tiene una varianza alta	222
Ejercicios	223

Capítulo 7. Ensamblaje y *random forests* 225

Clasificadores de votación.....	225
<i>Bagging</i> y <i>pasting</i>	229
<i>Bagging</i> y <i>pasting</i> en Scikit-Learn	230
Evaluación fuera de la bolsa.....	232
Parches aleatorios y subespacios aleatorios	233
<i>Random forests</i>	233
<i>Extra-Trees</i>	234
Importancia de las características	235
<i>Boosting</i>	235
AdaBoost	236
<i>Gradient boosting</i>	239
<i>Gradient boosting</i> basado en histogramas	243
<i>Stacking</i>	244
Ejercicios	247

Capítulo 8. Reducción de dimensionalidad 249

La maldición de la dimensionalidad	250
Enfoques principales para la reducción de dimensionalidad.....	251
Proyección	251
Aprendizaje de variedades.....	253
PCA	255
Preservar la varianza	255
Componentes principales	256
Proyección para bajar a d dimensiones	257
Utilizar Scikit-Learn	258
Ratio de varianza explicada	258
Elegir el número adecuado de dimensiones.....	258
PCA para compresión.....	260
PCA aleatorizado	261
PCA gradual.....	262
Proyección aleatoria	263
LLE	265
Otras técnicas de reducción de dimensionalidad.....	267
Ejercicios	268

Capítulo 9. Técnicas de aprendizaje no supervisado 271

Algoritmos de agrupamiento: k-medias y DBSCAN	272
K-medias.....	274
Límites de k-medias	284

Utilizar agrupamiento para la segmentación de imágenes.....	284
Utilizar agrupamiento para aprendizaje semisupervisado	286
DBSCAN	290
Otros algoritmos de agrupamiento	292
Mezclas gaussianas.....	294
Utilizar mezclas gaussianas para la detección de anomalías	298
Seleccionar el número de grupos.....	299
Modelos bayesianos de mezcla gaussiana	302
Otros algoritmos para detección de anomalías y de novedades	303
Ejercicios	304

Parte II. Redes neuronales y deep learning 307

Capítulo 10. Introducción a las redes neuronales artificiales con Keras 309

De las neuronas biológicas a las artificiales.....	310
Neuronas biológicas.....	311
Cálculos lógicos con neuronas	312
El perceptrón.....	313
El perceptrón multicapa y la retropropagación.....	317
PMC de regresión	321
PMC de clasificación	323
Implementación de PMC con Keras.....	325
Creación de un clasificador de imágenes utilizando la API secuencial	325
Creación de un PMC de regresión con la API secuencial	335
Creación de modelos complejos con la API funcional	336
Utilización de la API de subclasificación para crear modelos dinámicos.....	342
Guardar y restaurar un modelo	343
Utilización de retrollamadas	344
Uso de TensorBoard para visualización	346
Ajuste de los hiperparámetros de una red neuronal	349
Número de capas ocultas	354
Número de neuronas por capa oculta.....	355
Tasa de aprendizaje, tamaño de lote y otros hiperparámetros.....	356
Ejercicios	358

Capítulo 11. Entrenar redes neuronales profundas

361

Los problemas de desvanecimiento/explosión de gradientes	362
Inicialización de Glorot y He	363
Funciones de activación mejores	365
Normalización de lotes.....	371
Recorte de gradiente	376
Reutilizar redes preentrenadas	377
Aprendizaje por transferencia con Keras	378
Preentrenamiento no supervisado.....	380
Preentrenar con una tarea auxiliar	381
Optimizadores más rápidos.....	382
<i>Momentum</i>	382
Gradiente acelerado de Nesterov	384
AdaGrad.....	385
RMSProp	386
Adam.....	387
AdaMax.....	388
Nadam	388
AdamW	389
Programación de la tasa de aprendizaje.....	390
Evitar el sobreajuste mediante la regularización	395
Regularizaciones ℓ_1 y ℓ_2	395
Dropout	396
Monte Carlo (MC) Dropout.....	399
Regularización <i>max-norm</i>	402
Resumen y directrices prácticas.....	403
Ejercicios	404

Capítulo 12. Modelos personalizados y entrenamiento con TensorFlow

407

Un tour rápido por TensorFlow	407
Utilizar TensorFlow como NumPy	411
Tensores y operaciones.....	411
Tensores y NumPy	413
Conversiones de tipos	413
Variables	414
Otras estructuras de datos	414
Personalizar modelos y algoritmos de entrenamiento.....	415
Funciones de pérdida personalizadas.....	415

Guardar y cargar modelos que contengan componentes personalizados	416
Funciones de activación, inicializadores, regularizadores y restricciones personalizados	418
Métricas personalizadas	420
Capas personalizadas	422
Modelos personalizados	425
Pérdidas y métricas basadas en componentes internos del modelo....	427
Calcular gradientes utilizando diferenciación automática.....	429
Bucles de entrenamiento personalizados	432
Grafos y funciones de TensorFlow	435
AutoGraph y trazado	437
Reglas de las funciones TF.....	439
Ejercicios	440

Capítulo 13. Cargar y preprocesar datos con TensorFlow 443

La API tf.data.....	444
Encadenar transformaciones	445
Mezclar los datos	447
Preprocesar los datos.....	450
Juntarlo todo.....	451
Precarga.....	451
Utilizar el conjunto de datos con Keras	454
El formato TFRecord	455
Archivos TFRecord comprimidos	456
Breve introducción a los búferes de protocolo.....	456
Protobufs de TensorFlow	458
Cargar y analizar ejemplos.....	459
Manejar listas de listas usando el protobuf SequenceExample	460
Capas de preprocesamiento de Keras	461
La capa Normalization	462
La capa Discretization	464
La capa CategoryEncoding	465
La capa StringLookup	466
La capa Hashing.....	467
Codificar características categóricas utilizando <i>embeddings</i>	468
Preprocesamiento de texto.....	472
Utilizar componentes de modelos de lenguaje preentrenados	474
Capas de preprocesamiento de imágenes	475
El proyecto TensorFlow Datasets	476
Ejercicios	477

Capítulo 14. *Deep learning* para visión por ordenador usando redes neuronales convolucionales 479

La arquitectura de la corteza visual	480
Capas convolucionales.....	481
Filtros	483
Apilar múltiples mapas de características	485
Implementar capas convolucionales con Keras.....	487
Requisitos de memoria	490
Capas de <i>pooling</i>	491
Implementación de capas de <i>pooling</i> con Keras	493
Arquitecturas de RNC	495
LeNet-5	498
AlexNet	499
GoogLeNet	502
VGGNet.....	505
ResNet	505
Xception.....	509
SENet.....	510
Otras arquitecturas destacables.....	512
Elegir la arquitectura de RNC adecuada.....	514
Implementar una RNC ResNet-34 usando Keras	515
Utilizar modelos preentrenados desde Keras	516
Modelos preentrenados para aprendizaje por transferencia.....	518
Clasificación y localización	521
Detección de objetos.....	522
Redes completamente convolucionales	525
<i>You Only Look Once</i>	527
Seguimiento de objetos	530
Segmentación semántica	531
Ejercicios	535

Capítulo 15. Procesar secuencias utilizando RNR y RNC 537

Capas y neuronas recurrentes	538
Celdas de memoria.....	540
Secuencias de entrada y salida	541
Entrenamiento de RNR	542
Predicción de una serie temporal.....	543
La familia de modelos ARMA	549
Preparar los datos para modelos de <i>machine learning</i>	552
Realizar un pronóstico utilizando un modelo lineal	555

Realizar un pronóstico utilizando una RNR simple	556
Realizar un pronóstico utilizando una RNR profunda.....	557
Predecir series temporales multivariantes	559
Predecir varios pasos de tiempo más adelante	560
Realizar un pronóstico utilizando un modelo secuencia a secuencia.....	562
Manejar secuencias largas	565
Enfrentarse al problema de los gradientes inestables.....	565
Enfrentarse al problema de la memoria a corto plazo.....	568
Ejercicios	576

Capítulo 16. Procesamiento de lenguaje natural con RNR y atención

577

Generar texto shakespeariano utilizando una RNR a nivel de carácter	578
Crear el conjunto de datos de entrenamiento.....	579
Crear y entrenar el modelo de RNR a nivel de carácter.....	581
Generar texto shakespeariano falso	582
RNR con estado	584
Análisis de sentimiento	587
Enmascaramiento.....	590
Reutilización de <i>embeddings</i> y modelos de lenguaje preentrenados ...	593
Una red codificador-descodificador para la traducción automática neuronal.....	595
RNR bidireccionales	601
Haz local (<i>beam search</i>).....	603
Mecanismos de atención	605
Solo necesitas atención: la arquitectura Transformer original	609
Una avalancha de modelos de transformador	620
Transformadores de visión	624
Biblioteca Transformers de Hugging Face	629
Ejercicios	633

Capítulo 17. Autocodificadores, GAN y modelos de difusión

635

Representaciones de datos eficientes	637
Realizar PCA con un autocodificador lineal incompleto.....	639
Autocodificadores apilados	640
Implementación de un autocodificador apilado usando Keras	641
Visualizar las reconstrucciones.....	642
Visualizar el conjunto de datos Fashion MNIST	643

Preentrenamiento no supervisado utilizando autocodificadores	
apilados.....	644
Atar pesos	645
Entrenar autocodificadores de uno en uno	646
Autocodificadores convolucionales	648
Autocodificador con eliminación de ruido.....	649
Autocodificadores dispersos	651
Autocodificadores variacionales	654
Generar imágenes de Fashion MNIST	658
Redes generativas antagónicas.....	659
Las dificultades de entrenar GAN	663
GAN convolucionales profundas	665
Crecimiento progresivo de GAN	668
StyleGAN	671
Modelos de difusión	673
Ejercicios	681

Capítulo 18. Aprendizaje por refuerzo

683

Aprender a optimizar recompensas	684
Búsqueda de políticas	685
Introducción a OpenAI Gym	687
Políticas de redes neuronales	691
Evaluar acciones: el problema de la asignación de crédito.....	693
Gradientes de política	694
Procesos de decisión de Markov	699
Aprendizaje por diferencia temporal	703
Aprendizaje Q-Learning	704
Políticas de exploración.....	706
Q-Learning aproximado y Q-Learning profundo	707
Implementación de Q-Learning profundo	708
Variantes del Q-Learning profundo	713
Objetivos de valores Q fijos.....	713
Double DQN.....	714
Repetición de experiencia priorizada.....	714
Dueling DQN.....	715
Visión general de algunos algoritmos de aprendizaje por refuerzo	
populares.....	716
Ejercicios	720

Capítulo 19. Entrenar y desplegar modelos de TensorFlow a escala 721

Servir un modelo de TensorFlow	722
Utilizar TensorFlow Serving.....	722
Crear un servicio de predicción en Vertex AI	731
Realizar tareas de predicción por lotes en Vertex AI.....	738
Desplegar un modelo en un dispositivo móvil o integrado	740
Ejecutar un modelo en una página web.....	743
Uso de GPU para acelerar la computación.....	745
Conseguir tu propia GPU	746
Gestionar la RAM de la GPU.....	748
Colocar operaciones y variables en dispositivos	751
Ejecución paralela en múltiples dispositivos.....	752
Entrenar modelos en múltiples dispositivos	755
Paralelismo del modelo.....	755
Paralelismo de datos	758
Entrenar y escalar con la API de estrategia de distribución.....	764
Entrenar un modelo en un clúster de TensorFlow.....	765
Ejecutar trabajos de entrenamiento grandes en Vertex AI.....	769
Ajuste de hiperparámetros en Vertex AI	771
Ejercicios	775
¡Gracias!	776

Parte III. Apéndices 777

Apéndice A. Lista de comprobación de proyectos de machine learning 779

Enmarcar el problema y tener una visión amplia.....	779
Obtener los datos	780
Explorar los datos	780
Preparar los datos	781
Seleccionar modelos prometedores.....	782
Perfeccionar el sistema.....	782
Presentar la solución.....	783
¡Lanzar!	783

Apéndice B. Diferenciación automática 785

Diferenciación manual	785
Aproximación mediante diferencias finitas.....	786
Diferencia automática hacia delante	787
Diferenciación automática inversa.....	790

Apéndice C. Estructuras de datos especiales 793

Cadenas	793
Tensoros irregulares	794
Tensoros dispersos.....	795
Matrices tensoriales	796
Conjuntos	797
Colas	797

Apéndice D. Grafos de TensorFlow 799

Funciones TF y funciones concretas.....	799
Explorar definiciones y grafos de función	801
Atención al trazado	802
Utilizar AutoGraph para capturar estructuras de control	804
Manejar variables y otros recursos en funciones TF	805
Utilizar funciones TF con Keras (o no)	806

Índice alfabético 809

- TD-Gammon, 683
- teacher forcing*, 595
- técnica de sobremuestreo de minorías sintéticas (SMOTE), 500
- técnica *softmax* muestreada, 600
- temperatura, modelo Char-RNN, 583
- TensorBoard, 346-349, 410, 592, 769
- tensores, 411-413
 - de cadena, 415
 - de máscara, 590
 - dispersos, 795
 - simbólicos, 438, 800
- TensorFlow, 22, 22, 407-478, 721-775
 - arquitectura, 409
 - con NumPy, 411-415
 - conversiones de tipos, 413
 - crear función de entrenamiento, 454
 - desplegar modelo en un dispositivo móvil, 740-743
 - estructuras de datos especiales, 793-798
 - funciones y grafos, 799-807
 - gestión de GPU con, 748-751, 752-755
 - grafos y funciones, 408, 435-440, 799-807
 - hub.KerasLayer, 474
 - operaciones matemáticas, 412
 - operaciones y tensores, 408, 411-413
 - página, ejecutar modelo en, 744
 - plataformas y API disponibles, 409
 - variables, 414
- TensorFlow,
 - clúster de, 766-769
 - Datasets (TFDS), proyecto, 476-478
 - Extended (TFX), 410
 - Hub, 410, 474, 594
 - Lite, 410
 - playground, 324
 - Serving (TF Serving), 722-740
 - API gRPC, consultar a través de, 728
 - API REST, consultar a través de, 727
 - contenedor Docker, 726
 - crear servicio de predicciones, 731-738
 - desplegar nueva versión de modelo, 729-731
 - exportar SavedModels, 723-725
 - instalar e iniciar, 725-727
 - trabajos de predicciones de lotes en Vertex AI, 738
 - Text, 473, 589
- TensorFlow.js (TFJS) JavaScript, biblioteca, 744
- teorema “No hay almuerzo gratis”, 63
- teoría
 - de la información, 189, 216
 - hebbiana, 315
- término de suavizamiento, 372, 385
- tf.data, API, 444-455
 - encadenar transformaciones, 445-447
 - intercalar líneas de múltiples archivos, 448-449
 - mezclar datos, 447-448
 - precarga, 451-454
 - preprocesar los datos, 450-451
- tf.data.AUTOTUNE, 446
- tf.data.Dataset.from_tensor_slices(), 444-446
- tf.data.TFRecordDataset, 456, 459
 - usar conjunto de datos con Keras, 454-455
 - y capas de preprocesamiento de Keras, 463
- TFDS (TensorFlow Datasets), proyecto, 476-478
- TFJS (TensorFlow.js) JavaScript, biblioteca, 744
- TFLite, 741-743
- TFRecord, formato, 443, 455-461
- TFX (TensorFlow Extended), 410
- tiempo
 - polinomial, 216
 - real, 373
- Tikhonov, regularización de, 174-176, 179
- TNR (tasa de verdaderos negativos), 136
- Tokenizers, biblioteca, 589
- TPR (tasa de verdaderos positivos), 131, 136
- TPU (unidades de procesamiento tensorial), 408, 730, 743, 769
- traducción,
 - automática neuronal (NMT), 595-619
 - con transformadores, 620-624
 - y mecanismos de atención, 605-619
 - con RNR, 578, 595-604
- train_test_split(), 84, 113
- transferencia de estilo, GAN, 671
- transform(), 95, 98, 99, 104
- transformación de datos
 - transformadores
 - estimadores, 94
 - personalizados, 103-106
 - y escalado de características, 99-103
- transformaciones afines, 671
- transformador de logaritmos, 103
- transformadores
 - de imagen con eficiencia de datos (DeiT), 627
 - de visión (ViTs), 624-629

- multimodales, 627
- personalizados, 103-106
- TransformedTargetRegressor, 103
- Transformer, 609
- TransformerMixin, 104
- Transformers, biblioteca, 629-632
- transposición, operador de, 70
- truco
 - del *hash*, 467
 - kernel*, 198-202, 207-209
- true negatives, confusion matrix, 130
- trust region policy optimization* (TRPO), 718

U

- ULU (unidad lógica de umbral), 313, 317
- umbral
 - de decisión, 133-136
 - de densidad, 298
- unidad
 - de umbral lineal, 313
 - lineal exponencial (ELU), 367-370
 - lógica de umbral (ULU), 313, 317
- unidades
 - de activación cerradas, 574
 - de procesamiento tensorial (TPU), 408, 743, 769
- unidades residuales, 506
- Universal Sentence Encoder, 594-595

V

- validación cruzada, 62, 113-115, 120, 129-130, 171-174
 - de *k* iteraciones, 113, 129-130
- valor de estado óptimo, 701
- valores
 - atípicos, 272
 - estado-acción (valores Q), 702-703
 - Q, 702-703
- value_counts(), 78
- variables
 - colocar en GPU, 751
 - de holgura, 205
 - en TensorFlow, 414
 - manejar en funciones TF, 805-806
 - persistencia de, 422
- variacionales, autocodificadores, 654-658
- varianza
 - alta, con árboles de decisión, 222
 - compromiso sesgo-varianza, 174

- explicada, 258-260
 - trazar, 258-260
- preservar, 255
- varianza alta con árboles de decisión, 222
- vector
 - de parámetros, 153, 159, 183, 188
 - subgradiente, 179
- vectores,
 - columna, 153
 - de características, 71, 153, 204
 - normas para medir distancia, 71
 - soporte, 194
- ventaja de la acción, aprendizaje por refuerzo, 694
- verdaderos positivos, matriz de confusión, 130
- versión jerárquica de DBSCAN (HDBSCAN), 292
- Vertex AI, 121, 731-738, 769-771
- VGGNet, 505
- violaciones del margen, 195, 205
- Virtual, dispositivo GPU, 749
- visualización de datos, 37, 40-41, 86-92
 - árboles de decisión, 211-212
 - autocodificadores apilados, 642-643
 - ejercicio de principio a fin, 85-92
 - PMC con TensorBoard, 346-349
 - reducción de dimensionalidad, 250, 258
 - t-SNE, 267
- ViTs (transformadores de visión), 624-629
- votación, clasificadores de, 225-228

W

- WaveNet, 538, 574-575
- worker, tipo de tarea, 765
- workers, 759

X

- Xavier, inicialización, 363
- Xception (Extreme Inception), 509-510, 518-521
- XLA (álgebra lineal acelerada), 436
- XOR (exclusiva OR), problema, 316

Y

- You Only Look Once* (YOLO), 527-530

Z

- zero-shot*, aprendizaje, 622, 628