

# Contenidos

<b>Prefacio</b>	xxi
<hr/>	
<b>1. Dotar a los ordenadores de la capacidad de aprender de los datos</b>	<b>1</b>
<b>Creación de máquinas inteligentes para transformar los datos en conocimiento</b>	<b>2</b>
<b>Los tres tipos de aprendizaje automático</b>	<b>3</b>
Formulación de pronósticos sobre el futuro con el aprendizaje supervisado	• 3
<i>Clasificación para pronosticar etiquetas de clase</i>	• 4
<i>Regresión para el pronóstico de resultados continuos</i>	• 5
Resolución de problemas interactivos con el aprendizaje reforzado	• 7
Descubrimiento de estructuras ocultas con el aprendizaje no supervisado	• 8
<i>Detección de subgrupos mediante el agrupamiento (clustering)</i>	• 8
<i>Reducción de la dimensionalidad para comprimir los datos</i>	• 9
<b>Introducción a la terminología básica y la notación</b>	<b>10</b>
Notación y convenciones que se utilizan en el libro	• 10
Terminología del aprendizaje automático	• 12
<b>Hoja de ruta para crear sistemas de aprendizaje automático</b>	<b>13</b>
Preprocesamiento: cómo dar forma a los datos	• 14
Entrenamiento y selección de modelos predictivos	• 14
Evaluación de modelos y pronóstico de instancias de datos ocultos	• 15
<b>Utilización de Python en aprendizaje automático</b>	<b>16</b>
Instalación de Python y sus paquetes desde Python Package Index	• 16
Uso del gestor de paquetes y distribución de Anaconda Python	• 17
Paquetes para computación científica, ciencia de datos y aprendizaje automático	• 18
<b>Resumen</b>	<b>20</b>

<b>2. Entrenamiento de algoritmos sencillos de aprendizaje automático para tareas de clasificación</b> .....	<b>21</b>
Neuronas artificiales: un vistazo a la historia de los inicios del aprendizaje automático .....	22
Definición formal de neurona artificial • 23	
Regla de aprendizaje del perceptrón • 25	
<b>Implementación de algoritmos de aprendizaje del perceptrón en Python</b> .....	<b>28</b>
Una API del perceptrón orientada a objetos API • 28	
Entrenamiento de un modelo de perceptrón con el conjunto de datos Iris • 32	
<b>Neuronas lineales adaptativas y convergencia del aprendizaje</b> .....	<b>38</b>
Minimización de las funciones de pérdida con el descenso del gradiente • 39	
Implementación de Adaline en Python • 42	
Mejora del descenso del gradiente mediante el escalado de características • 46	
Aprendizaje automático a gran escala y descenso del gradiente estocástico • 49	
<b>Resumen</b> .....	<b>54</b>
<b>3. Un recorrido por los clasificadores de aprendizaje automático con Scikit-learn</b> .....	<b>57</b>
<b>Elección de un algoritmo de clasificación</b> .....	<b>57</b>
<b>Primeros pasos con Scikit-learn: entrenamiento del perceptrón</b> .....	<b>58</b>
<b>Modelización de las probabilidades de clase mediante regresión logística</b> .....	<b>64</b>
Regresión logística y probabilidades condicionales • 65	
Aprendizaje de los pesos del modelo mediante la función de pérdida logística • 69	
Conversión de una implementación de Adaline en un algoritmo de regresión logística • 72	
Entrenamiento de un modelo de regresión logística con Scikit-learn • 76	
Cómo abordar el sobreajuste utilizando la regularización • 79	
<b>Clasificación de margen máximo con máquinas de vectores de soporte</b> .....	<b>82</b>
Intuición de margen máximo • 83	
Tratamiento de un caso no separable linealmente mediante variables de holgura • 84	
Implementaciones alternativas en Scikit-learn • 86	
<b>Resolución de problemas no lineales utilizando el kernel en SVM</b> .....	<b>86</b>
Métodos del kernel para datos no separables linealmente • 86	
Uso del truco del kernel para encontrar hiperplanos de separación en un espacio de alta dimensión • 89	
<b>Aprendizaje utilizando árboles de decisión</b> .....	<b>92</b>

Maximización de IG: cómo sacar el máximo partido a su dinero • 93	
Creación de un árbol de decisión • 98	
Combinación de múltiples árboles de decisión mediante bosques aleatorios • 100	
<b>K vecinos más cercanos: un algoritmo de aprendizaje perezoso</b> .....	104
<b>Resumen</b> .....	108
<b>4. Elaboración de conjuntos de datos de entrenamiento adecuados:</b>	
<b>preprocesamiento</b> .....	109
<b>Tratamiento de los datos que faltan</b> .....	109
Identificación de los valores que faltan en las tablas de datos • 110	
Eliminación de ejemplos de entrenamiento o de características con valores perdidos • 111	
Atribución de valores perdidos • 112	
Interpretación de la API del estimador de Scikit-learn • 114	
<b>Tratamiento de datos categóricos</b> .....	115
Codificación de datos categóricos con Pandas • 115	
Asignación de las características ordinales • 116	
Codificación de las etiquetas de clase • 117	
Codificación one-hot de las características nominales • 118	
<i>Opcional: codificación de características ordinales • 121</i>	
<b>Partición del conjunto de datos en conjuntos de entrenamiento y de prueba separados</b> .....	122
<b>Las características deben estar en la misma escala</b> .....	125
<b>Selección de características significativas</b> .....	128
Regularizaciones L1 y L2 como penalizaciones contra la complejidad del modelo • 128	
Interpretación geométrica de la regularización L2 • 129	
Soluciones dispersas con la regularización L1 • 130	
Algoritmos de selección secuencial de características • 134	
<b>Evaluación de la importancia de las características con bosques aleatorios</b> ....	141
<b>Resumen</b> .....	143
<b>5. Compresión de datos mediante la reducción de la dimensionalidad</b> ..	145
<b>Reducción de la dimensionalidad no supervisada mediante el análisis de componentes principales</b> .....	146
Etapas más importantes del análisis de componentes principales • 146	
Extracción de los componentes principales paso a paso • 148	

Varianzas total y explicada • 151	
Transformación de características • 153	
Análisis de componentes principales en Scikit-learn • 156	
Evaluación de las contribuciones de las características • 159	
<b>Compresión de datos supervisados mediante el análisis discriminante lineal</b>	<b>162</b>
Análisis de componentes principales frente al análisis discriminante lineal • 162	
Funcionamiento interno del análisis discriminante lineal • 163	
Cálculo de matrices de dispersión • 164	
Selección de discriminantes lineales para el nuevo espacio de características • 166	
Proyección de ejemplos en el nuevo espacio de características • 169	
LDA mediante Scikit-learn • 170	
<b>Reducción de la dimensionalidad no lineal y visualización de datos</b>	<b>172</b>
¿Por qué considerar la reducción de la dimensionalidad no lineal? • 172	
Visualización de datos mediante la incorporación de vecinos estocásticos distribuidos en $t$ • 174	
<b>Resumen</b>	<b>177</b>
<b>6. Aprendizaje de las mejores prácticas para la evaluación de modelos y el ajuste de hiperparámetros</b>	<b>179</b>
<b>Optimización de flujos de trabajo con pipelines</b>	<b>179</b>
Carga del conjunto de datos Breast Cancer Wisconsin • 180	
Combinación de transformadores y estimadores en una pipeline • 182	
<b>Uso de la validación cruzada de <math>k</math> iteraciones (<math>k</math> subconjuntos) para evaluar el rendimiento del modelo</b>	<b>184</b>
El método de retención • 184	
Validación cruzada de $k$ iteraciones • 185	
<b>Depuración de algoritmos con curvas de aprendizaje y validación</b>	<b>189</b>
Diagnóstico de problemas de sesgo y varianza con curvas de aprendizaje • 190	
Como abordar el sobreajuste y el infraajuste con curvas de validación • 193	
<b>Ajuste de los modelos de aprendizaje automático mediante la búsqueda en la red</b>	<b>195</b>
Ajuste de hiperparámetros mediante la búsqueda en la red • 195	
Configuraciones de los hiperparámetros con la búsqueda aleatoria • 197	
Búsqueda de hiperparámetros más eficiente en recursos con reducción sucesiva a la mitad • 200	
Selección de algoritmos con validación cruzada anidada • 202	

---

<b>Observación de diferentes métricas de evaluación del rendimiento .....</b>	<b>204</b>
Lectura de la matriz de confusión • 204	
Optimización de la precisión y la recuperación de un modelo de clasificación • 206	
Representación de la característica operativa del receptor • 209	
Métricas de puntuación para la clasificación multiclase • 211	
Cómo hacer frente al desequilibrio de clases • 212	
<b>Resumen .....</b>	<b>215</b>
<b>7. Combinación de diferentes modelos para el aprendizaje de conjunto ....</b>	<b>217</b>
<b>Aprendizaje de conjunto .....</b>	<b>217</b>
<b>Combinación de clasificadores mediante el voto mayoritario .....</b>	<b>221</b>
Implementación de un clasificador de voto mayoritario sencillo • 221	
Utilización del principio de votación por mayoría para hacer pronósticos • 226	
Evaluación y ajuste del clasificador conjunto • 229	
<b>Bagging: elaboración de un conjunto de clasificadores a partir de muestras bootstrap .....</b>	<b>235</b>
Bagging en pocas palabras • 236	
Aplicación de bagging para clasificar ejemplos en el conjunto de datos Wine • 237	
<b>Aprovechamiento de los alumnos débiles mediante el refuerzo adaptativo .....</b>	<b>241</b>
Cómo funciona el refuerzo adaptativo • 242	
Aplicación de AdaBoost con Scikit-learn • 246	
<b>Refuerzo del gradiente: entrenamiento de un conjunto basado en los gradientes de pérdida .....</b>	<b>250</b>
Comparación de AdaBoost con el refuerzo del gradiente • 250	
Esquema del algoritmo general de refuerzo del gradiente • 251	
Explicación del algoritmo de refuerzo del gradiente para la clasificación • 253	
Ilustración del refuerzo del gradiente para la clasificación • 254	
Utilización de XGBoost • 257	
<b>Resumen .....</b>	<b>259</b>
<b>8. Aplicación del aprendizaje automático al análisis de opiniones .....</b>	<b>261</b>
<b>Preparación de datos de reseñas de películas de IMDb para el tratamiento de textos .....</b>	<b>262</b>
Obtención del conjunto de datos de reseñas de películas • 262	
Preprocesamiento del conjunto de datos de películas en un formato más conveniente • 263	

<b>Presentación del modelo bolsa de palabras</b> .....	265
Transformación de palabras en vectores de características • 265	
Evaluación de la relevancia de las palabras mediante la frecuencia de los términos-frecuencia inversa de los documentos • 267	
Limpieza de datos de texto • 270	
Procesamiento de documentos en tokens • 272	
<b>Entrenamiento de un modelo de regresión logística para la clasificación de documentos</b> .....	274
<b>Trabajo con conjuntos de datos más grandes: algoritmos en línea y aprendizaje fuera del núcleo</b> .....	277
<b>Modelado de temas con asignación latente de Dirichlet</b> .....	281
Descomposición de documentos de texto con LDA • 282	
LDA con Scikit-learn • 282	
<b>Resumen</b> .....	286
<b>9. Pronóstico de variables objetivo continuas con análisis de regresión ...</b>	<b>289</b>
<b>Introducción a la regresión lineal</b> .....	290
Regresión lineal simple • 290	
Regresión lineal múltiple • 291	
<b>Exploración del conjunto de datos Ames Housing</b> .....	292
Carga del conjunto de datos Ames Housing en un DataFrame • 292	
Visualización de las características importantes de un conjunto de datos • 294	
Observaciones de las relaciones mediante la matriz de correlación • 296	
<b>Implementación del modelo de regresión lineal mediante mínimos cuadrados ordinarios</b> .....	298
Resolución de la regresión para los parámetros de regresión con el descenso del gradiente • 299	
Estimación del coeficiente del modelo de regresión mediante Scikit-learn • 303	
<b>Ajuste del modelo de regresión robusto mediante RANSAC</b> .....	306
<b>Evaluación del rendimiento de los modelos de regresión lineal</b> .....	309
<b>Uso de métodos regularizados para la regresión</b> .....	314
<b>Conversión del modelo de regresión lineal en una curva: regresión polinómica</b> .....	316
Adición de términos polinómicos usando Scikit-learn • 316	
Modelado de relaciones no lineales en el conjunto de datos Ames Housing • 318	
<b>Tratamiento de relaciones no lineales mediante bosques aleatorios</b> .....	321
Regresión de árbol de decisión • 321	

---

Regresión de bosque aleatorio • 323	
<b>Resumen</b> .....	<b>326</b>
<b>10. El trabajo con datos no etiquetados: análisis de clustering</b> .....	<b>327</b>
<b>Agrupación de objetos por similitud mediante k-means</b> .....	<b>328</b>
Clustering de k-means con Scikit-learn • 328	
Una forma más inteligente de colocar los centroides iniciales de los clústeres utilizando k-means++ • 333	
Clustering duro frente a clustering blando • 334	
Uso del método del codo para encontrar el número óptimo de clústeres • 336	
Cuantificación de la calidad del clustering mediante gráficos de silueta • 337	
<b>Organización de clústeres en forma de árbol jerárquico</b> .....	<b>342</b>
Agrupación de clústeres de forma ascendente • 343	
Realización de la agrupación jerárquica en la matriz de distancias • 344	
Anexión de dendrogramas a un mapa de calor • 348	
Aplicación de clustering aglomerativo a través de Scikit-learn • 350	
<b>Localización de regiones de alta densidad mediante DBSCAN</b> .....	<b>351</b>
<b>Resumen</b> .....	<b>356</b>
<b>11. Implementación de una red neuronal artificial multicapa desde cero</b> ....	<b>359</b>
<b>Modelado de funciones complejas con redes neuronales artificiales</b> .....	<b>359</b>
Resumen de redes neuronales de una capa • 361	
Introducción a la arquitectura de redes neuronales multicapa • 363	
Activación de una red neuronal por propagación hacia delante • 365	
<b>Clasificación de dígitos escritos a mano</b> .....	<b>368</b>
Obtención y preparación del conjunto de datos MNIST • 368	
Implementación del perceptrón multicapa • 372	
Codificación del bucle de entrenamiento de la red neuronal • 377	
Evaluación del rendimiento de la red neuronal • 382	
<b>Entrenamiento de redes neuronales artificiales</b> .....	<b>386</b>
Cálculo de la función de pérdida • 386	
Desarrollo de conocimiento de la retropropagación • 388	
Entrenamiento de redes neuronales por retropropagación • 390	
<b>Sobre la convergencia en redes neuronales</b> .....	<b>394</b>
<b>Unas últimas palabras sobre la implementación de redes neuronales</b> .....	<b>395</b>
<b>Resumen</b> .....	<b>396</b>

<b>12. Paralelización de redes neuronales con PyTorch .....</b>	<b>397</b>
<b>PyTorch y el rendimiento del entrenamiento .....</b>	<b>397</b>
Retos del rendimiento • 398	
¿Qué es PyTorch? • 399	
Cómo vamos a aprender PyTorch • 400	
<b>Primeros pasos con PyTorch .....</b>	<b>400</b>
Instalación de PyTorch • 401	
Creación de tensores en PyTorch • 402	
Manipulación del tipo de datos y la forma de tensores • 403	
Aplicación de operaciones matemáticas a tensores • 404	
División, apilamiento y concatenación de tensores • 405	
<b>Elaboración de pipelines de entrada en PyTorch .....</b>	<b>407</b>
Creación de PyTorch DataLoader a partir de tensores existentes • 408	
Combinación de dos tensores en un conjunto de datos mixto • 409	
Barajar, agrupar y repetir • 410	
Creación de conjuntos de datos a partir de archivos del disco de almacenamiento local • 412	
Obtención de los conjuntos de datos disponibles en la biblioteca torchvision.datasets • 416	
<b>Elaboración de modelos de NN en PyTorch .....</b>	<b>420</b>
Módulo de redes neuronales de PyTorch (torch.nn) • 420	
Elaboración de un modelo de regresión lineal • 421	
Entrenamiento de modelos mediante los módulos torch.nn y torch.optim • 425	
Elaboración de un perceptrón multicapa para clasificar flores en el conjunto de datos Iris • 426	
Evaluación del modelo entrenado en el conjunto de datos de prueba • 429	
Almacenamiento y recarga del modelo entrenado • 430	
<b>Elección de funciones de activación para redes neuronales multicapa .....</b>	<b>431</b>
Recapitulación de la función logística • 431	
Estimación de las probabilidades de clase en la clasificación multiclase mediante la función softmax • 433	
Ampliación del espectro de salida mediante la tangente hiperbólica • 434	
Activación de la unidad lineal rectificada • 434	
<b>Resumen .....</b>	<b>438</b>



---

<b>13. Profundización: la mecánica de PyTorch</b> .....	<b>441</b>
Características esenciales de PyTorch .....	442
Grafos de computación de PyTorch .....	443
Comprensión de los grafos de computación • 443	
Creación de grafos en PyTorch • 443	
Objetos tensoriales PyTorch para almacenar y actualizar los parámetros del modelo .....	444
Cálculo de gradientes mediante diferenciación automática .....	447
Cálculo de gradientes de pérdida con respecto a las variables entrenables • 447	
Comprensión de la diferenciación automática • 449	
Ejemplos adversarios • 449	
Simplificación de las implementaciones de arquitecturas más frecuentes mediante el módulo <code>torch.nn</code> .....	449
Implementación de modelos basados en <code>nn.Sequential</code> • 450	
Elección de la función de pérdida • 451	
Resolución de problemas de clasificación XOR • 452	
Cómo flexibilizar la elaboración de modelos con <code>nn.Module</code> • 457	
Cómo escribir capas personalizadas en PyTorch • 459	
Proyecto uno: pronóstico de la eficiencia de combustibles de coches .....	464
Trabajo a realizar con columnas de características • 464	
Entrenamiento de modelos de regresión DNN • 468	
Proyecto dos: clasificación de dígitos escritos a mano del MNIST .....	470
Las API de PyTorch de alto nivel: una breve introducción a PyTorch-Lightning .....	473
Configuración del modelo PyTorch Lightning • 474	
Configuración de los cargadores de datos para Lightning • 477	
Entrenamiento de modelos con la clase PyTorch Lightning Trainer • 479	
Evaluación del modelo con TensorBoard • 479	
Resumen .....	483
<b>14. Clasificación de imágenes con redes neuronales   convolucionales profundas</b> .....	<b>485</b>
Componentes de las CNN .....	485
Comprensión de las CNN y jerarquías de características • 486	
Realización de convoluciones discretas • 488	
<i>Convoluciones discretas en una dimensión</i> • 489	

<i>Entradas de relleno para controlar el tamaño de los mapas de características de salida</i> • 491	
<i>Determinación del tamaño de la salida de la convolución</i> • 493	
<i>Realización de convoluciones discretas en 2D</i> • 494	
Capas de submuestreo • 498	
<b>Poner todo junto: implementación de CNN</b> .....	<b>499</b>
Trabajo con varios canales de entrada o de color • 500	
Regularización de NN con regularización y abandono de L2 • 503	
Funciones de pérdida para la clasificación • 507	
<b>Implementación de CNN profundas utilizando PyTorch</b> .....	<b>509</b>
La arquitectura de CNN multicapa • 509	
Carga y preprocesamiento de datos • 510	
Implementación de CNN usando el módulo torch.nn • 512	
<i>Configuración de capas de CNN en PyTorch</i> • 512	
<i>Construcción de CNN en PyTorch</i> • 513	
<b>Clasificación de sonrisas a partir de imágenes faciales mediante CNN</b> .....	<b>519</b>
Carga del conjunto de datos CelebA • 519	
Transformación de imágenes e incremento de datos • 520	
Entrenamiento de un clasificador de sonrisas CNN • 527	
<b>Resumen</b> .....	<b>534</b>
<b>15. Modelado de datos secuenciales mediante redes neuronales recurrentes</b> .....	<b>537</b>
<b>Introducción a los datos secuenciales</b> .....	<b>537</b>
Modelado de datos secuenciales: el orden importa • 538	
Datos secuenciales frente a datos de series temporales • 538	
Representación de secuencias • 539	
Diferentes categorías del modelado de secuencias • 539	
<b>RNN para modelar secuencias</b> .....	<b>541</b>
Comprender el flujo de datos en las RNN • 541	
Cálculo de activaciones en RNN • 543	
Recurrencia oculta frente a recurrencia de salida • 545	
Retos del aprendizaje de las interacciones de largo alcance • 548	
Células de memoria a corto y largo plazo • 550	
<b>Implementación de RNN para el modelado de secuencias en PyTorch</b> .....	<b>552</b>
Proyecto uno: pronóstico de las opiniones en las reseñas de cine de IMDb • 552	

<i>Preparación de los datos de reseñas de películas</i> • 553	
<i>Capas de incorporación (embedding) para la codificación de frases</i> • 558	
<i>Elaboración de modelos RNN</i> • 560	
<i>Elaboración de un modelo RNN para la tarea de análisis de opiniones</i> • 561	
Proyecto dos: modelado lingüístico a nivel de caracteres en PyTorch • 566	
<i>Preprocesamiento del conjunto de datos</i> • 567	
<i>Elaboración de un modelo RNN a nivel de carácter</i> • 572	
<i>Fase de evaluación: generación de nuevos pasajes de texto</i> • 574	
<b>Resumen</b> .....	<b>579</b>
<b>16. Transformadores: mejora del procesamiento del lenguaje natural con mecanismos de atención</b> .....	<b>581</b>
<b>Adición de un mecanismo de atención a las RNN</b> .....	<b>582</b>
El mecanismo de atención ayuda a las RNN a acceder a la información • 582	
Mecanismo de atención original para las RNN • 583	
Procesamiento de entradas mediante RNN bidireccional • 584	
Generación de salidas a partir de vectores de contexto • 585	
Cálculo de los pesos de atención • 586	
<b>Introducción al mecanismo de autoatención</b> .....	<b>587</b>
Empezando por una forma básica de autoatención • 587	
Parameterización del mecanismo de autoatención: atención del producto punto escalado • 591	
<b>La atención es todo lo que necesitamos: introducción a la arquitectura original del transformador</b> .....	<b>595</b>
Codificación de las incorporaciones de contexto mediante atención multicabezal • 597	
Aprendizaje de un modelo lingüístico: decodificador y atención multicabezal enmascarada • 601	
Detalles de la implementación: codificaciones posicionales y normalización de capas • 602	
<b>Elaboración de modelos lingüísticos a gran escala aprovechando los datos no etiquetados</b> .....	<b>605</b>
Preentrenamiento y ajuste de los modelos de transformadores • 605	
Aprovechamiento de datos no etiquetados con GPT • 608	
Utilización de GPT-2 para generar texto nuevo • 612	
Preentrenamiento bidireccional con BERT • 614	
Lo mejor de ambos mundos: BART • 618	

<b>Reajuste de modelos BERT en PyTorch</b> .....	621
Carga del conjunto de datos de reseñas de películas de IMDb • 622	
Tokenización del conjunto de datos • 624	
Carga y reajuste de modelos BERT preentrenados • 625	
Cómo reajustar los transformadores de forma más cómoda mediante la API Trainer • 630	
<b>Resumen</b> .....	634
<b>17. Redes generativas antagónicas con las que sintetizar nuevos datos</b> ....	637
<b>Introducción a las redes generativas antagónicas</b> .....	637
Empezando por los autocodificadores • 638	
Modelos generativos para sintetizar nuevos datos • 640	
Generación de nuevas muestras con GAN • 642	
Comprensión de las funciones de pérdida de las redes generadoras y discriminadoras en un modelo GAN • 643	
<b>Implementación de un modelo GAN desde cero</b> .....	645
Entrenamiento de modelos GAN en Google Colab • 646	
Implementación de las redes generadora y discriminadora • 649	
Definición del conjunto de datos de entrenamiento • 652	
Entrenamiento de modelos GAN • 654	
<b>Mejora de la calidad de las imágenes sintetizadas mediante GAN convolucionales y de Wasserstein</b> .....	661
Convolución traspuesta • 661	
Normalización por lotes • 663	
Implementación del generador y del discriminador • 666	
Medidas de disimilitud entre dos distribuciones • 673	
Uso de la distancia EM en la práctica para las GAN • 677	
Penalización del gradiente • 677	
Implementación de WGAN-GP para entrenar modelos DCGAN • 678	
Colapso de modo • 683	
<b>Otras aplicaciones de las GAN</b> .....	684
<b>Resumen</b> .....	685
<b>18. Redes neuronales de grafos para captar las dependencias en datos estructurados en grafos</b> .....	687
<b>Introducción a los datos de grafos</b> .....	688
Grafos no dirigidos • 688	

---

Grafos dirigidos • 689	
Grafos etiquetados • 690	
Representación de las moléculas como grafos • 690	
<b>Comprensión de las convoluciones de grafos</b> .....	<b>690</b>
Motivaciones para utilizar las convoluciones de grafos • 691	
Implementación de una convolución básica de grafos • 693	
<b>Implementación de GNN en PyTorch desde cero</b> .....	<b>698</b>
Definición del modelo NodeNetwork • 698	
Codificación de la capa de convolución del grafo de NodeNetwork • 700	
Preparación de DataLoader • 704	
Uso de NodeNetwork para hacer pronósticos • 707	
<b>Implementación de GNN con la biblioteca PyTorch Geometric</b> .....	<b>708</b>
<b>Otras capas de GNN y desarrollos recientes</b> .....	<b>714</b>
Convoluciones de grafos espectrales • 714	
Reducción (pooling) • 716	
Normalización • 718	
Referencias a las redes neuronales de grafos en la literatura avanzada • 719	
<b>Resumen</b> .....	<b>721</b>
<b>19. Aprendizaje reforzado para la toma de decisiones en entornos complejos</b> .....	<b>723</b>
<b>Introducción: aprender de la experiencia</b> .....	<b>724</b>
Comprensión del aprendizaje reforzado • 724	
Definición de la interfaz agente-entorno de los sistemas de aprendizaje reforzado • 726	
<b>Fundamentos teóricos de RL</b> .....	<b>727</b>
Procesos de decisión de Markov • 727	
<i>Formulación de los procesos de decisión de Markov</i> • 728	
<i>Visualización del proceso de Markov</i> • 730	
Tareas episódicas o continuas • 731	
Terminología RL: retorno, política y función de valor • 731	
<i>Retorno</i> • 732	
<i>Política</i> • 733	
<i>Función de valor</i> • 734	
Programación dinámica mediante la ecuación de Bellman • 735	

<b>Algoritmos de aprendizaje reforzado .....</b>	<b>736</b>
Programación dinámica • 737	
<i>Evaluación de políticas: pronóstico de la función de valor con programación dinámica</i> • 738	
<i>Mejora de la política mediante la función de valor estimada</i> • 738	
<i>Iteración de la política</i> • 739	
<i>Iteración del valor</i> • 739	
Aprendizaje reforzado con Monte Carlo • 740	
<i>Estimación de la función de valor del estado mediante MC</i> • 741	
<i>Estimación de la función de valor de la acción mediante MC</i> • 741	
<i>Consecución de la política óptima mediante el control de MC</i> • 741	
<i>Mejora de la política: cálculo de la política codiciosa a partir de la función de valor de la acción</i> • 742	
Aprendizaje por diferencia temporal • 742	
<i>Pronóstico TD</i> • 743	
<i>Control de TD en la política (SARSA)</i> • 744	
<i>Control de TD fuera de la política (Q-learning)</i> • 744	
<b>Implementación de nuestro primer algoritmo RL .....</b>	<b>745</b>
Presentación del kit de herramientas OpenAI Gym • 745	
<i>El trabajo con los entornos existentes en OpenAI Gym</i> • 746	
<i>Ejemplo del tipo mundo cuadriculado</i> • 748	
<i>Implementación del entorno de mundo cuadriculado en OpenAI Gym</i> • 748	
Resolución del problema de mundo cuadriculado con Q-learning • 755	
<b>Un vistazo a Q-learning profundo .....</b>	<b>760</b>
Entrenamiento de modelos DQN según el algoritmo Q-learning • 761	
<i>Memoria de repetición</i> • 761	
<i>Determinación de los valores objetivo para el cálculo de la pérdida</i> • 762	
Implementación del algoritmo Q-learning profundo • 764	
<b>Resumen del capítulo y del libro .....</b>	<b>768</b>