

1 Ante una nueva tecnología disruptiva

Se está considerando la inteligencia artificial como la nueva revolución industrial, corazón de lo que algunos llaman industria 4.0. Pues bien, el *Deep Learning* es el motor de este proceso y en los siguientes capítulos hablaremos extensamente de ello. Pero en este capítulo vamos primero a situar el tema, ver porque la inteligencia artificial está ya aquí y porqué ha venido para quedarse.

1.1 La inteligencia artificial está cambiando nuestras vidas

Nos encontramos ante vertiginosos avances en la calidad y prestaciones de una amplia gama de tecnologías cotidianas: en el caso del reconocimiento de voz, la transcripción de voz a texto ha experimentado avances increíbles, y ya está disponible en diferentes dispositivos. Estamos interactuando cada vez más con nuestros ordenadores simplemente hablando con ellos.

También ha habido avances espectaculares en el procesamiento del lenguaje natural. Por ejemplo, simplemente haciendo clic en el símbolo de micro de *Google Translate*, el sistema transcribirá a otro idioma lo que está dictando. *Google Translate* ya permite convertir oraciones de una lengua a otra en 32 pares de idiomas, y ofrece traducción de texto para más de 100.

A su vez, los avances en la visión por computador también son enormes: ahora nuestros ordenadores, por ejemplo, pueden reconocer imágenes y generar descripciones textuales de su contenido en segundos.

Estas tres áreas son cruciales para dar rienda suelta a las mejoras en robótica, drones o automóviles sin conductor, estando la inteligencia artificial en el corazón de toda esta innovación tecnológica, que últimamente avanza tan rápidamente gracias al *Deep Learning*.

Y todo ello a pesar de que la inteligencia artificial todavía no se ha desplegado

ampliamente y es difícil hacerse una idea del gran impacto que tendrá, al igual que en 1995 lo era el imaginarse el impacto futuro de internet. En aquel entonces, la mayoría de la gente no veía cómo internet era relevante para ellos y cómo iba a cambiar sus vidas.

Personas como Sundar Pichai, CEO de Google dicen que “el impacto de la inteligencia artificial en la historia de la humanidad es comparable con la electricidad y el fuego⁵”. Para él, la inteligencia artificial es una de las cosas más importantes en las que la humanidad está trabajando y que al igual que la gente aprendió a utilizar el fuego para los beneficios de la humanidad, también necesitó superar sus desventajas.

Quiero creerme que Pichai es muy optimista respecto a la inteligencia artificial y que está convencido que podría usarse para ayudar a resolver algunos de los retos que tenemos la humanidad encima de la mesa. Quizás esta comparativa es una exageración, eso solo lo sabremos con el tiempo; pero yo de ustedes le tendría puesto el ojo a la inteligencia artificial, porque algo está cambiando, y a todos nos conviene estar atentos a lo que se avecina.

1.2 Inteligencia artificial, Machine Learning y Deep Learning

Creo que antes de continuar estaría bien que concretáramos un poco que entendemos por inteligencia artificial, *Machine Learning* y *Deep Learning*, tres términos que aparecerán muy a menudo a lo largo del libro.

⁵ El impacto de la AI en la historia de la humanidad es comparable con la electricidad y el fuego. Blog. [online] Disponible en: <http://jorditorres.org/inteligencia-artificial-electricidad-fuego/> [Accedido: 18/03/2018]

Inteligencia artificial

¿A qué nos referimos cuando hablamos de inteligencia artificial? Una extensa y precisa definición (y descripción de sus ámbitos) se encuentra en el libro de Stuart Rusell⁶ y Peter Norvig⁷ titulado *Artificial Intelligence, a modern approach*⁸, la obra de texto sobre inteligencia artificial más popular en el mundo universitario y, sin duda para mí, el mejor punto de partida para tener una visión global del tema. Pero intentando hacer una aproximación más generalista (propósito de este libro) podríamos aceptar una definición simple en la que por inteligencia artificial nos referimos a aquella inteligencia que muestran las máquinas, en contraste con la inteligencia natural de los humanos. En este sentido, una posible definición concisa y general de inteligencia artificial podría ser el esfuerzo para automatizar tareas intelectuales normalmente realizadas por humanos.

Como tal, el área de inteligencia artificial (*Artificial Intelligence* en inglés) es un campo muy amplio que abarca muchas áreas de conocimiento relacionadas con el aprendizaje automático; incluso se incluyen muchos más enfoques no siempre catalogados como aprendizaje automático por mis colegas universitarios expertos en el tema. Además, a lo largo del tiempo, a medida que los computadores han sido cada vez más capaces de “hacer cosas”, se han ido cambiando las tareas o tecnologías consideradas como “inteligentes”.

Esto se explica porque desde los años 50, la inteligencia artificial ha experimentado varias oleadas de optimismo, seguidas por la decepción y la pérdida de financiación e interés (épocas conocidas como *AI winter*⁹), seguidas de nuevos enfoques, éxito y financiación. Además, durante la mayor parte de su

⁶ Stuart J. Russell. Wikipedia. [online]. Disponible en: https://en.wikipedia.org/wiki/Stuart_J._Russell [Accedido: 16/04/2018]

⁷ Peter Norvig Wikipedia. [online]. Disponible en: https://en.wikipedia.org/wiki/Peter_Norvig [Accedido: 16/04/2018]

⁸ Artificial Intelligence: A Modern Approach (AIMA) · 3rd edition, Stuart J Russell and Peter Norvig, Prentice hall, 2009. ISBN 0-13-604259-7

⁹ AI winter. Wikipedia. [online]. Disponible en: https://en.wikipedia.org/wiki/AI_winter [Accedido: 16/04/2018]

historia, la investigación en inteligencia artificial se ha dividido en subcampos basados en consideraciones técnicas o herramientas matemáticas concretas y con comunidades de investigación que no se comunicaban suficientemente entre sí.

Por tanto, de momento creo que podemos avanzar sin una definición muy detallada de inteligencia artificial: es suficiente para este libro el darse cuenta de que está aquí y ya nos acompaña con todo lo que estamos haciendo.

Machine Learning

Como decíamos en el anterior apartado, avances como el reconocimiento de voz, el procesado de lenguaje natural o la visión por computador son cruciales para desencadenar mejoras en robótica, drones, coches que se conducen solos, entre muchas otras áreas que están cambiando el futuro próximo. Muchos de estos avances han sido posibles gracias a una familia de técnicas conocida popularmente como *Deep Learning* del que hablaremos extensamente. Pero antes creo que es interesante para hacernos una imagen global correcta especificar que el *Deep Learning* es una subparte de una de las áreas de la inteligencia artificial conocida como *Machine Learning*.

El *Machine Learning*, en general traducido al castellano como “aprendizaje automático” (aunque yo voy a mantener su nombre en inglés en este libro), es en sí mismo un gran campo de investigación y desarrollo. En concreto, el *Machine Learning* se podría definir como el subcampo de la inteligencia artificial que proporciona a los ordenadores la capacidad de aprender sin ser explícitamente programados, es decir, sin que necesiten que el programador indique explícitamente las reglas que debe seguir para lograr su tarea sino que las hace automáticamente.

Generalizando, podemos decir que el *Machine Learning* consiste en desarrollar para cada problema, un “algoritmo” de predicción para un caso de uso particular. Estos algoritmos aprenden de los datos con el fin de encontrar

patrones o tendencias para comprender *qué nos dicen* los datos y de esta manera construir un modelo para predecir y clasificar los elementos.

Dada la madurez del área de investigación en *Machine Learning*, existen muchos enfoques bien establecidos para el aprendizaje automático por parte de máquinas. Cada uno de ellos utiliza una estructura algorítmica diferente para optimizar las predicciones basadas en los datos recibidos. El *Machine Learning*, es un amplio campo con una compleja taxonomía de algoritmos que se agrupan en general en tres grandes categorías: aprendizaje supervisado, aprendizaje no supervisado y *Reinforcement Learning*.

Nos referimos a que el “aprendizaje es supervisado” cuando los datos que usamos para el entrenamiento incluyen la solución deseada, llamada “etiqueta” (*label*). Algunos de los algoritmos más populares de *Machine Learning* en esta categoría son la regresión lineal, la regresión logística, *support vector machines*, *decision trees*, *random forest* y redes neuronales.

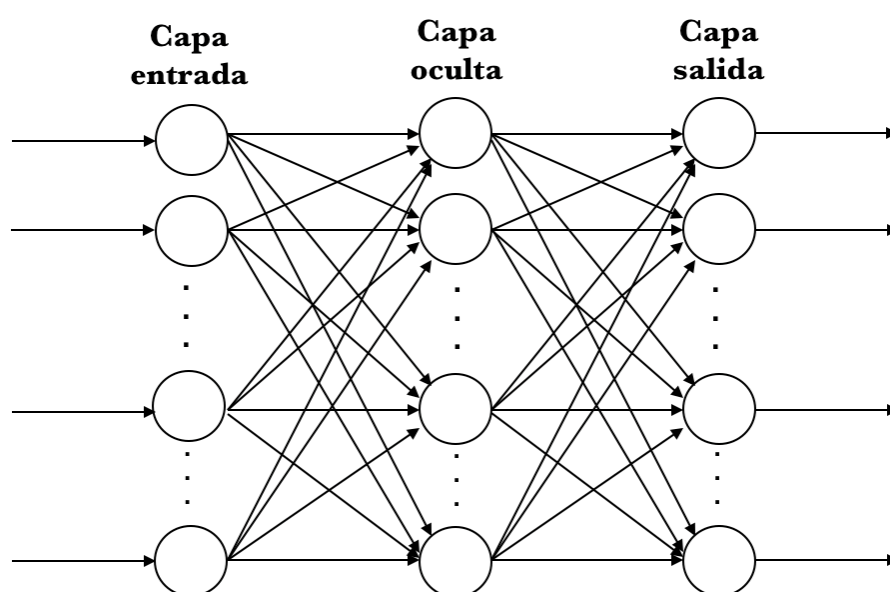
En cambio, cuando nos referimos a un “aprendizaje no supervisado” los datos de entrenamiento no incluyen las etiquetas, y será el algoritmo el que intentará clasificar la información por sí mismo. Algunos de los algoritmos más conocidos de esta categoría son *clustering (K-means)* o *principal component analysis (PCA)*.

También hablamos de *Reinforcement Learning* (o aprendizaje por refuerzo, traducción de algunos autores) cuando el modelo se implementa en forma de un agente que deberá explorar un espacio desconocido y determinar las acciones a llevar a cabo mediante prueba y error: aprenderá por sí mismo gracias a las recompensas y penalizaciones que obtiene de sus acciones. El agente debe crear la mejor estrategia posible (políticas) para obtener la mayor recompensa en tiempo y forma. Este aprendizaje permite ser combinado con otros tipos, y está ahora mismo muy de moda puesto que el mundo real presenta muchos de estos escenarios.

Redes neuronales artificiales y Deep Learning

Un caso especial de algoritmos de *Machine Learning* son las redes neuronales artificiales. Si les ayuda, para visualizar su estructura, pueden considerar que los algoritmos son similares a las neuronas humanas y su capacidad para la obtención de resultados, como habrán oído en alguna ocasión, aunque personalmente creo que poco tiene que ver.

En el caso concreto del *Deep Learning* (que en castellano se traduce a veces como aprendizaje profundo, aunque usaré la versión en inglés), las estructuras algorítmicas antes mencionadas permiten modelos que están compuestos de múltiples capas de procesamiento para aprender representaciones de datos, con múltiples niveles de abstracción que representan una serie de transformaciones lineales y no lineales con el objetivo de encontrar los parámetros óptimos para transformar las entradas de tal manera que se aproximen al máximo a las salidas esperadas. Una aproximación gráfica simple a una red neuronal *Deep Learning* es



En concreto, aquí representamos una red neuronal artificial con 3 capas, con una capa de entrada (*Input Layer*) que recibe los datos de entrada y con una capa de salida (*Output Layer*) que devuelve la predicción realizada. Las capas que tenemos en medio se llaman *Hidden Layers* y podemos tener muchas, cada una con distinta cantidad de neuronas. Veremos más adelante que las neuronas, representadas por los círculos, estarán interconectadas unas con otras de diferente manera entre las neuronas de las distintas capas.

En general, hoy en día estamos manejando redes neuronales artificiales con muchísimas capas, que literalmente están apiladas una encima de la otra; de aquí el concepto de *deep* (profundidad de la red), donde cada una de ellas está a su vez compuesta por muchísimas neuronas, cada una con sus parámetros que, a su vez, realizan una transformación simple de los datos que reciben de neuronas de la capa anterior para pasarlos a las de la capa posterior. La unión de todas permite descubrir patrones complejos.

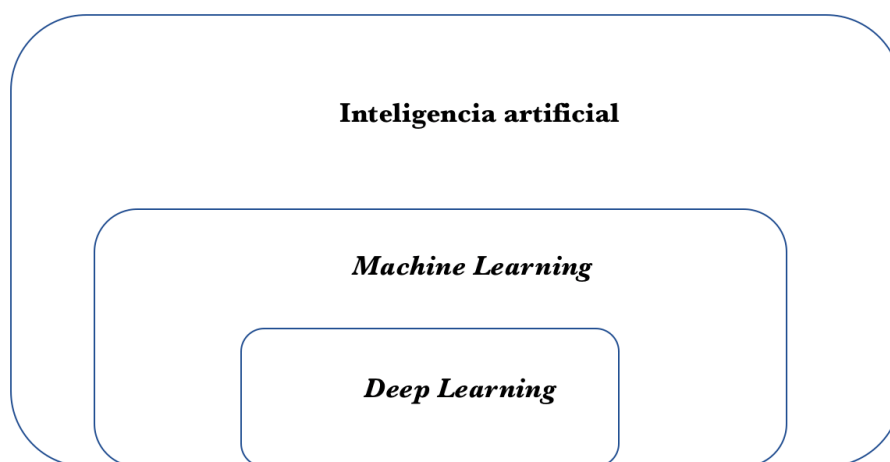
Como veremos en detalle más adelante, los avances en *Deep Learning* han mejorado drásticamente el estado de la técnica en reconocimiento de voz, reconocimiento de objetos visuales, detección de objetos y muchos otros dominios, siendo una de las técnicas que han puesto la inteligencia artificial en el foco de interés de las empresas y de aquí el gran interés que ahora mismo suscitan.

Pero aunque el *Deep Learning* a menudo se presenta envuelto en una cierta mística, con referencias a algoritmos que “funcionan como el cerebro”, que “piensan” o “entienden”, a mi entender la realidad aún dista bastante de este sueño de ciencia ficción. Además, creo que sus conceptos básicos pueden ser explicados de manera relativamente fácil a lectores con una base de conocimiento en informática y sobretodo ganas de aprender, siendo este mi propósito.

Antes de acabar, me gustaría dar una magnitud del problema que conlleva programar en estos momentos los algoritmos de *Deep Learning*: diferentes capas sirven para diferentes propósitos, y cada parámetro e hiperparámetro importa

mucho en el resultado final; esto lo hace extremadamente complicado a la hora de intentar afinar la programación, pareciendo más un arte que una ciencia para los que se adentran por primera vez en el área. Pero esto no implica que sea algo misterioso, si bien es cierto que queda mucho por investigar, sino que simplemente hace falta muchas horas de aprendizaje y muchas horas de prácticas.

La siguiente figura, que uso en mis transparencias de clase, resume visualmente la idea intuitiva de que el *Deep Learning* es solo una parte de la inteligencia artificial, aunque en estos momentos quizás es la más dinámica y la está haciendo realmente vibrar. Y de la misma manera que antes les mencionaba el libro de Stuart Rusell y Peter Novig como libro base de inteligencia artificial, para *Deep Learning* nos encontramos con un excelente libro, titulado *DEEP LEARNING*¹⁰, realizado por Ian Goodfellow, Yoshua Bengio y Aaron Corville que es el campo base en estos momentos para el aprendizaje del tema en más profundidad.



¹⁰ Deep Learning. I Goodfellow, Y. Bengio and A Corville. MIT Press 2016. Disponible también en acceso libre en <http://www.deeplearning.org> [consulta: 20/01/2018].

1.3 ¿Por qué ahora?

En tan solo diez años¹¹, cuatro de las cinco empresas más grandes del mundo por capitalización de mercado han cambiado: Exxon Mobil, General Electric, Citigroup y Shell Oil están fuera y Apple, Alphabet (la compañía matriz de Google), Amazon y Facebook han tomado su lugar. Solo Microsoft mantiene su posición. Ya se han percatado que todas ellas dominan la nueva era digital en que nos encontramos inmersos. Estamos hablando de empresas que basan su poderío en inteligencia artificial en general, y en particular *Deep Learning*.

John McCarthy acuñó el término inteligencia artificial en la década de los 50 y fue uno de los padres fundadores de la inteligencia artificial junto con Marvin Minsky. También en 1958 Frank Rosenblatt construyó un prototipo de red neuronal, que llamó el *Perceptron*. Además, las ideas clave de las redes neuronales *Deep Learning* para la visión por computador ya se conocían en 1989; también los algoritmos fundamentales de *Deep Learning* para series temporales como LSTM (que trataremos más adelante), ya fueron desarrollados en 1997, por poner algunos ejemplos. Entonces, ¿por qué este *boom* de la inteligencia artificial?

Sin duda, se ha debido a varios factores que vamos a comentar a continuación, pero claramente algunos destacan. A modo general, podríamos resumir que los cuellos de botella más importantes a lo largo de las dos últimas décadas para el despliegue de estas tecnologías fueron la gran cantidad de datos que se requieren y la ingente capacidad de computación requerida. Sin embargo, otros factores han contribuido a desencadenar el potencial de la inteligencia artificial y las tecnologías relacionadas. A continuación vamos a hablar de algunos de ellos.

¹¹ Los datos que aparecen en esta sección son los disponibles en el momento de escribirla a principios del año 2018.

Los datos, el combustible para la inteligencia artificial

La inteligencia artificial requiere grandes conjuntos de datos para el entrenamiento de sus modelos aunque, afortunadamente, la creación y disponibilidad de datos ha crecido exponencialmente. El progreso exponencial en el *hardware* de almacenamiento de los últimos años, asociado a los espectaculares avances en técnica para su gestión con bases de datos NoSQL¹², han permitido disponer de enormes conjuntos de datos para entrenar a los modelos de inteligencia artificial.

Más allá de los aumentos en la disponibilidad de datos que ha propiciado internet y sus múltiples aplicaciones, los recursos de datos especializados han catalizado el progreso del área. Muchas bases de datos abiertas han apoyado el rápido desarrollo de algoritmos de inteligencia artificial. Un ejemplo ImageNet¹³, una base de datos disponible libremente con más de 10 millones de imágenes etiquetadas a mano. Pero lo que hace ImageNet especial no es precisamente su tamaño, sino la competición que anualmente realiza, siendo una excelente manera de motivar a investigadores e ingenieros.

Mientras que en los primeros años las propuestas tradicionales de reconocimiento de imágenes se basaban en algoritmos de visión por computador, en el 2012 Alex Krizhevsky usó una red neuronal *Deep Learning*, ahora conocida por AlexNet, que redujo el ratio de error a más de la mitad de lo que se estaba consiguiendo por aquel entonces. Ya en el 2015, el algoritmo ganador rivalizó con las capacidades humanas, y a día de hoy los algoritmos de *Deep Learning* superan con creces los ratios de error en esta competición de los que tienen los humanos.

Pero no solo ImageNet es una de las bases de datos disponible que se han usado

¹² Wikipedia, NoSQL. [online]. Disponible en: <https://es.wikipedia.org/wiki/NoSQL> [Accedido: 15/04/2018]

¹³ The ImageNet Large Scale Visual Recognition Challenge (ILSVRC). [online]. Disponible en: www.image-net.org/challenges/LSVRC. [Accedido: 12/03/2018]

para entrenar redes *Deep Learning* durante estos últimos años, muchas otras han sido populares, como MNIST¹⁴, CIFAR¹⁵, SVHN¹⁶, STL¹⁷ o IMDB¹⁸. Hablaremos de ellas más adelante. También es importante mencionar aquí Kaggle¹⁹, una plataforma que aloja competiciones de análisis de datos donde compañías e investigadores aportan sus datos mientras ingenieros de datos de todo el mundo compiten por crear los mejores modelos de predicción o clasificación.

Nuevo hardware masivamente paralelo

Debido a que la investigación en este campo del *Deep Learning* ha estado guiada por los hallazgos experimentales más que por la teoría, estos recientes avances espectaculares solo han sido posibles cuando los datos y el *hardware* apropiados han estado disponibles para probar nuevas ideas o ampliar las viejas.

Gracias a la ley de Moore²⁰, hoy en día podemos resolver problemas que hubieran sido intratables hace solo pocos años debido a la capacidad de computación. Por ejemplo, el computador al que yo tenía acceso en el año 1982, donde ejecuté mi primer programa con tarjetas perforadas, era un Fujitsu que permitía ejecutar entre un millón o dos millones de operaciones por segundo. 30 años después, en el 2012, el supercomputador Marenstrum era solo 1.000.000.000 veces más rápido :-).

En el 2012, el incremento de capacidad de computación que cada año conseguíamos de los ordenadores era gracias a la mejora de la CPU. Sin

¹⁴ MNIST [online]. Disponible en: <http://yann.lecun.com/exdb/mnist/> [Accedido: 12/03/2018]

¹⁵ CIFAR [online]. Disponible en: <http://www.cs.toronto.edu/~kriz/cifar.html> [Accedido: 12/03/2018]

¹⁶ SVHN [online]. Disponible en: <http://ufldl.stanford.edu/housenumbers/> [Accedido: 12/03/2018]

¹⁷ STL [online]. Disponible en: <http://ai.stanford.edu/~acoates/stl10/> [Accedido: 12/03/2018]

¹⁸ IMDB [online]. Disponible en: <https://www.kaggle.com/pankrzysiu/keras-imdb-reviews> [Accedido: 12/03/2018]

¹⁹ Kaggle [online]. Disponible en: <http://www.kaggle.com> [Accedido: 12/03/2018]

²⁰ Ley de Moore. Wikipedia. [online]. Disponible en: https://es.wikipedia.org/wiki/Ley_de_Moore [Accedido: 12/03/2018]

embargo, desde entonces el incremento de capacidad de computación para el *Deep Learning* no ha sido solo gracias a ello, sino también a los nuevos sistemas masivamente paralelos basados en unidades de procesamiento gráfico (*graphical processing units*) conocidos popularmente por GPU²¹, que resultan decenas de veces más eficientes que las CPU tradicionales.

Las GPU se desarrollaron originalmente para acelerar el juego 3D que requiere el uso repetido de un proceso matemático como un cálculo de la matriz. Originalmente, compañías como Nvidia y AMD desarrollaron estos chips rápidos y masivamente paralelos para tarjetas gráficas dedicadas a videojuegos.

Se vio que las GPUs útiles para juegos 3D eran muy adecuadas también para acelerar cálculos sobre matrices numéricas; por ello, este *hardware* en realidad permitía beneficiar a la comunidad científica, y en el 2007 Nvidia lanzó el lenguaje de programación CUDA para poder programar sus GPUs. Centros como el BSC empezó a usar *clusters* de GPUs para aplicaciones numéricas altamente paralelizables.

Pero como veremos, las redes neuronales artificiales básicamente realizan operaciones matriciales que son también altamente paralelizables. Y esto es lo que hizo en 2012 el equipo de Alex Krizhevsky, que entrenó su algoritmo *Deep Learning* AlexNet con GPUs. Desde entonces se empezaron a usar las GPUs para esta competición, y en estos momentos todos los grupos que investigan en *Deep Learning* están usando este *hardware* o alternativas equivalentes que han salido recientemente.

Hoy en día, la industria *Deep Learning* requiere ir más allá de las GPUs y por ello se están diseñando chips de procesado especialmente pensados para algoritmos *Deep Learning*. Por ejemplo, en 2016 Google anunció que había construido un procesador dedicado para la inferencia de aprendizaje profundo llamado la

²¹ Wikipedia. Unidad de procesamiento gráfico o GPU. [online] Disponible en: https://es.wikipedia.org/wiki/Unidad_de_procesamiento_gráfico [Accedido: 12/02/2018]

Tensor Processing Unit (TPU)²². El TPU proporciona un alto rendimiento en la inferencia de aprendizaje profundo, mientras que el consumo era de solo una fracción de la potencia de otros procesadores. Desde entonces han salido al mercado varias propuestas.

Democratización de la computación

Ahora bien, ¿qué pasa si uno no dispone de esta capacidad de computación en su empresa? La inteligencia artificial hasta ahora ha sido principalmente el juguete de las grandes compañías de tecnología como Amazon, Baidu, Google o Microsoft, así como algunas nuevas empresas que disponían de estas capacidades. Para muchos otros negocios y partes de la economía, los sistemas de inteligencia artificial hasta ahora han sido demasiado costosos y demasiado difíciles de implementar por completo. Estamos hablando del *Cloud Computing*²³.

Pero ahora estamos entrando en otra era de democratización de la computación, y las empresas pueden disponer de acceso a grandes *data centers* de más de 28.000 metros cuadrados (cuatro veces el campo del Barça), con cientos de miles de servidores dentro.

El *Cloud Computing* ha revolucionado la industria mediante la democratización de la computación y ha cambiado completamente la manera de operar de los negocios. Y ahora es el turno de cambiar el escenario de la inteligencia artificial y el *Deep Learning*, ofreciendo una gran oportunidad para las pequeñas y medianas empresas que no pueden construir este tipo de infraestructuras, pero en cambio el *Cloud Computing* sí se lo puede ofrecer; de hecho, ofrece acceso a una capacidad de computación que antes solo estaba disponible para grandes organizaciones o

²² Wikipedia. Tensor Processing Unit. [online]. Disponible en: https://en.wikipedia.org/wiki/Tensor_processing_unit [Accedido: 20/04/2018]

²³ Empresas en la nube: ventajas y retos del Cloud Computing. Jordi Torres. Editorial Libros de Cabecera. 2011.

gobiernos.

Además, los proveedores de *Cloud* están ahora ofreciendo lo que se conoce como *Artificial Intelligence algorithms as a Service (AI-as-a-Service)*, servicios de inteligencia artificial a través del *Cloud* que pueden entrelazarse y trabajar conjuntamente con aplicaciones internas de las empresas a través de simples APIs REST²⁴.

Esto implica que está al alcance de casi todos, ya que se trata de un servicio que solo se paga por el tiempo utilizado. Esto es disruptivo, porque ahora mismo permite a los desarrolladores de *software* usar y poner en producción prácticamente cualquier algoritmos de inteligencia artificial en producción en un santiamén.

Amazon, Microsoft, Google e IBM están liderando esta oleada de servicios AIaaS que permiten desde entrenamientos a puestas en producción de manera rápida. En el momento de escribir este libro, Amazon AIaaS estaba disponible a dos niveles: analíticas predictivas con *Amazon Machine Learning*²⁵ y la herramienta SageMaker²⁶ para la construcción y despliegue rápido de modelos. Microsoft ofrece sus servicios a través de su *Azure Machine Learning* que puede ser dividido en dos categorías principales también: *Azure Machine Learning Studio*²⁷ y *Azure Intelligence Gallery*²⁸. Google ofrece *Prediction API*²⁹ y el *Google ML Engine*³⁰. IBM ofrece servicios AIaaS a través de su *Watson Analytics*³¹. Y además no olvidemos

²⁴ Wikipedia. REST. [online]. Disponible en:

https://en.wikipedia.org/wiki/Representational_state_transfer [Accedido: 12/03/2018]

²⁵ Amazon ML [online]. Disponible en: <https://aws.amazon.com/aml/> [Accedido: 12/03/2018]

²⁶ SageMaker [online]. Disponible en: <https://aws.amazon.com/sagemaker/> [Accedido: 12/03/2018]

²⁷ Azure ML Studio [online]. Disponible en: <https://azure.microsoft.com/en-us/services/machine-learning-studio/> [Accedido: 12/03/2018]

²⁸ Azure Intelligent Gallery [online]. Disponible en: <https://gallery.azure.ai> [Accedido: 12/03/2018]

²⁹ Google Prediction API [online]. Disponible en: <https://cloud.google.com/prediction/docs/> [Accedido: 12/03/2018]

³⁰ Google ML engine [online]. Disponible en: <https://cloud.google.com/ml-engine/docs/technical-overview> [Accedido: 12/03/2018]

³¹ Watson Analytics [online]. Disponible en: <https://www.ibm.com/watson/> [Accedido: 12/03/2018]

soluciones que ya vienen de startups, como PredicSis³² and BigML³³.

Sin duda, la inteligencia artificial liderará la próxima revolución. Su éxito dependerá en gran medida de la creatividad de las empresas y no tanto de la tecnología *hardware* en parte gracias al *Cloud Computing*.

Un mundo open-source para la comunidad *Deep Learning*

Hace algunos años, el *Deep Learning* requería experiencia en lenguajes como C++ y CUDA. Hoy en día, con habilidades básicas de Python hay suficientes. Esto ha sido posible gracias al gran número de *frameworks* de *software* de código abierto que han ido apareciendo, como Keras, central en nuestro libro. Estos *frameworks* facilitan enormemente la creación y entrenamiento de los modelos y permiten abstraer al diseñador del algoritmo las peculiaridades del *hardware* para acelerar los procesos de entrenamiento.

Puestos a destacar algunos, les propongo que se queden con TensorFlow, Keras y PyTorch, pues son los más dinámicos en estos momentos si nos basamos en los *contributors* y *commits* o *starts* de estos proyectos en Github³⁴.

En concreto, recientemente ha tomado mucho impulso TensorFlow³⁵ y sin duda es el dominante. Fue originalmente desarrollado por investigadores e ingenieros del grupo de Google Brain en Google. El sistema fue diseñado para facilitar la investigación en *Machine Learning* y hacer más rápido la transición de un prototipo de investigación a un sistema de producción. Si miramos en su página de Github³⁶ veremos que tenemos, a fecha de escribir este libro, más de 30.000 *commits*, más

³² PredicSis [online]. Disponible en: <https://predicSis.ai> [Accedido: 12/03/2018]

³³ BigML [online]. Disponible en: <https://bigml.com> [Accedido: 12/03/2018]

³⁴ Véase <https://www.kdnuggets.com/2018/02/top-20-python-ai-machine-learning-open-source-projects.html>

³⁵ Véase <http://tensorflow.org>

³⁶ Véase <https://github.com/tensorflow/tensorflow>

de 1.400 *contributors* y casi 100.000 *starts*. Nada desdeñable.

Le sigue Keras³⁷ con una API de alto nivel para redes neuronales, que lo convierte en el entorno perfecto para iniciarse en el tema. El código se especifica en Python, y en estos momentos es capaz de ejecutarse encima de tres entornos destacados: TensorFlow, CNTK o Theano. En principio, el usuario puede cambiar el motor de ejecución sin cambiar su código de Keras. Ahora mismo, Keras cuenta con más de 4.500 *commits*, más de 650 *contributors* y más de 27.000 *starts*.

PyTorch y Torch³⁸ son dos entornos de *Machine Learning* implementados en C, usando OpenMP y CUDA para sacar provecho de infraestructuras altamente paralelas. PyTorch es la versión más focalizada para *Deep Learning* y basada en Python. Es un entorno popular en mi campo de investigación puesto que permite mucha flexibilidad en la construcción de las redes neuronales y tener tensores dinámicos entre otras cosas. En el momento de escribir este libro, Pytorch cuenta con más de 10.500 *commits*, alrededor de 600 *contributors* y más de 13.000 *starts*.

Finalmente, y aunque no es entorno exclusivo de *Deep Learning*, es importante mencionar Scikit-learn³⁹ que se usa muy a menudo en la comunidad de *Deep Learning* para el preprocesado de los datos⁴⁰. Scikit-learn cuenta con más de 22.500 *commits*, más de 1.000 *contributors* y más de 27.000 *starts*.

Pero como ya he avanzado, hay muchísimos otros *frameworks* orientados a *Deep Learning*. Los que destacaríamos son Theano⁴¹, Caffe⁴², Caffe2⁴³ (Facebook

³⁷ Véase <https://keras.io>

³⁸ Véase <http://pytorch.org>

³⁹ Véase <http://scikit-learn.org>

⁴⁰ Véase <http://scikit-learn.org/stable/modules/preprocessing.html>

⁴¹ Véase <http://deeplearning.net/software/theano>

⁴² Véase <http://caffe.berkeleyvision.org>

⁴³ Véase <https://caffe2.ai>

Research), CNTK⁴⁴ (Microsoft), MXNET⁴⁵ , Deeplearning4j⁴⁶, Chainer⁴⁷ , Kaldi⁴⁸, Lasagne⁴⁹, Leaf⁵⁰, MatConvNet⁵¹ y SoooA⁵² entre muchos otros.

Una cultura de publicación abierta

En estos últimos años, en esta área de investigación, en contraste con otras campos científicos, se ha generado una cultura de publicación abierta, en la que muchos investigadores publican sus resultados inmediatamente, sin esperar la aprobación de la revisión por pares habitual en los congresos, en bases de datos como por ejemplo la arxiv.org de la Universidad de Cornell (arXiv)⁵³. Esto conlleva que haya mucho *software* disponible en *open source* asociado a estos artículos que permiten que este campo de investigación se mueva tremendamente rápido, puesto que cualquier nuevo hallazgo está inmediatamente a disposición para toda la comunidad para verlo y, si es el caso, construir encima.

Esto supone una gran oportunidad para los usuarios de estas técnicas. El motivo para publicar sus últimos avances abiertamente por parte de los grupos de investigación pueden ser diversos; por ejemplo Google, al publicar los resultados, consolida su reputación como líder en el sector, atrayendo la siguiente ola de talento, que como comentaremos, es uno de los principales *stoppers* del tema.

⁴⁴ Véase <https://github.com/Microsoft/CNTK>

⁴⁵ Véase <https://mxnet.apache.org>

⁴⁶ Véase <https://deeplearning4j.org>

⁴⁷ Véase <https://chainer.org>

⁴⁸ Véase <http://kaldi-asr.org/doc/dnn.html>

⁴⁹ Véase <https://lasagne.readthedocs.io/en/latest/>

⁵⁰ Véase <https://github.com/autumnai/leaf>

⁵¹ Véase <http://www.vlfeat.org/matconvnet/>

⁵² Véase <https://github.com/laonbud/SoooA/>

⁵³ Véase <https://arxiv.org>

Mejoras en los algoritmos

Y gracias a la mejora del *hardware* y al disponer de más poder de computación por parte de los científicos que investigaban en algoritmos, se ha podido avanzar muchísimo en diseñar y probar nuevos algoritmos para poder superar importantes limitaciones de los algoritmos detectadas, y mejorar otras también importantes. Por ejemplo, hasta no hace muchos años era muy difícil entrenar redes de muchas capas desde un punto de vista del algoritmo. Pero en este último decenio ha habido impresionantes avances con mejoras en las funciones de activación, uso de redes preentrenadas, mejoras en algoritmos de optimización del entrenamiento, descubrir diversas técnicas que se aplican a mejorar el *backpropagation*, etc. Hoy, algorítmicamente hablando, podemos entrenar modelos de centenares de capas sin ningún problema. Otra cosa es si tenemos los recursos de computación para hacerlo realidad.

1.4 Preparar entorno de trabajo

Antes de continuar, les recomiendo que tengan el entorno de desarrollo instalado en su ordenador para ir probando lo que se va explicando en cada capítulo, pues recuerden el enfoque del libro de “*learn by doing*”.

Jupyter nootbook

Debido a que Keras es básicamente una librería de Python, requerimos hacer un uso completo del intérprete de Python. Nuestra propuesta es usar Jupyter⁵⁴, puesto que es un entorno de desarrollo muy extendido que permite sacar partido a la interactividad de Python y, a la vez, proporciona una gran versatilidad para compartir parte de códigos con anotaciones a través de la comodidad e

⁵⁴ Véase <https://jupyter.org>

independencia de plataforma que ofrece un navegador web.

Por este motivo, los *notebook* son usados a menudo para desarrollar redes neuronales en la comunidad de científicos e ingenieros de datos. Un *notebook* es un fichero generado por Jupyter Notebook o Jupyter Lab que se puede editar desde un navegador web, permitiendo mezclar la ejecución de código Python con anotaciones. Además, se puede ir ejecutándolo a porciones, ejecutadas independientemente, ofreciendo a los desarrolladores mucha interactividad.

Si no tienen instalado Jupyter pueden seguir las indicaciones de su página oficial⁵⁵. El código de este libro está disponible en forma de *notebooks* en el *github*⁵⁶ del libro, aunque este se puede ejecutar como un programa normal en Python si así lo desea el lector.

Keras

Para la instalación de la librería de Keras en un ordenador personal, el lector se puede dirigir a su página oficial, donde encontrará una versión actualizada de los pasos a seguir para la instalación de la última versión de la librería⁵⁷. En esta página podemos ver la indicación de tener previamente instalado uno de los *backend* que necesita, y nos recomienda usar TensorFlow (yo también se lo recomiendo). Para ello pueden visitar la página de instalación de Tensorflow⁵⁸ y seleccionar la versión de acuerdo a su sistema operativo. A continuación, para instalar Keras solo les queda seguir las instrucciones de su página de instalación.

Una instalación con solo una CPU nos será suficiente para seguir el libro. Pero como veremos a la que nos adentramos en redes neuronales convolucionales, nos puede ser útil tener potencia de cálculo basada en general en GPUs si no

⁵⁵ Véase <http://jupyter.org/install>

⁵⁶ Véase Github del libro <PENDIENTE>

⁵⁷ Véase <https://keras.io/#installation>

⁵⁸ Véase <https://www.tensorflow.org/install/>

queremos que nuestras ejecuciones para entrenar redes tarden muchas horas. En este caso, si el lector dispone de este *hardware*, en la misma página de instalación de Keras encontrará las instrucciones para la instalación de la versión de la librería para GPUs.

Docker

Una alternativa que les propongo es usar una imagen de *docker* que he preparado para una de mis asignaturas. *Docker*⁵⁹ es un proyecto de código abierto que se ha hecho muy popular entre la comunidad de programadores ya que permite automatizar el despliegue de programas usando contenedor software que permiten aislar las aplicaciones entre si. De esta manera podemos preparar *docker* con todo el software que necesitamos para ejecutar el código que usamos en este libro y simplificando el proceso de instalación. Si deciden ir por esta opción, les indico los pasos a seguir para poder usar la imagen del *docker* que tengo preparado para una de mis asignatura en la UPC.

Primero tenemos que instalar *docker* en nuestro ordenador. Para las plataformas Windows 10 Pro, Windows 10 Enterprise, MacOS o Linux podemos dirigirnos a *Docker Store*⁶⁰. Para otras versiones de Windows podemos dirigirse a *Docker Toolbox*⁶¹.

A continuación debemos descargar y ejecutar la imagen de *docker*. Para ello abriremos un *terminal* (en Mac/Linux), *cmd* o *powershell* (en Windows 10 Pro) o *Docker CLI* (para otras versiones de Windows) y luego podemos ejecutar:

```
docker pull jorditorresbcn/dl
```

⁵⁹ Véase <https://www.docker.com>

⁶⁰ Véase <https://store.docker.com/search?type=edition&offering=community>

⁶¹ Véase https://docs.docker.com/toolbox/toolbox_install_windows/

Los usuarios de MacOS y Windows tienen que tener el programa *docker* abierto para poder ejecutar el commando *docker*. Esta imagen está basada en Ubuntu 16.04 con el siguiente software instalado: Python3.5, Keras, TensorFlow, nano, htop, iPython, Jupyter, matplotlib, NLTK y git.

Para ejecutar la imagen de *docker* por primera vez deben abrir un nuevo terminal y ejecutar:

```
docker run -it -p 8888:8888 jorditorresbcn/dl:latest
```

En el caso que queramos usar una image del *docker* anteriormente usada y la queremos reabrir podemos hacerlo de esta manera:

```
docker ps -a
docker ps -a docker start -i <ID de su contenedor docker>
```

Para lanzar el Jupyter Notebook dentro del contenedor podemos ejecutar el siguiente comando:

```
jupyter notebook --ip=0.0.0.0 --allow-root
```

Podemos ignorar el error que puede aparecer “No web browser found: could not locate runnable browser”.

Ahora solo nos queda abrir en nuestro ordenador el navegador e ir a <http://localhost:8888>. El password es dl. En el caso de usuarios Windows que experimentan problemas de conectividad pueden visitar este [link](#)⁶².

Para asegurar que todo funciona correctamente les propongo que creen un nuevo notebook y ejecuten un código como el siguiente que imprime una gráfica

⁶² Véase https://github.com/jorditorresBCN/dlaimet/blob/master/windows_proxy.md

con 50 puntos aleatorios:

```
%matplotlib inline
from matplotlib import pyplot as plt
import numpy as np
N = 50
x = np.random.rand(N)
y = np.random.rand(N)
plt.scatter(x, y)
plt.show()
```

Cloud

Si queremos usar una GPU, y no disponemos de ello, podemos considerar usar los servicios que hay disponibles a través del *Cloud* que ofrecen diferentes proveedores. Pero dada la diversidad de plataformas y el cambio constante en las APIs que proporcionan los proveedores de *Cloud* en relación al acceso al *hardware* GPUs (que van incorporando a sus servicios últimamente), he considerado que era arriesgado aquí concretar más las indicaciones de los pasos a seguir para ejecutar un Jupyter en el *Cloud* sin el riesgo que las instrucciones indicadas queden obsoletas antes de incluso ver la luz el libro.

Por ello les propongo que en el momento de leer estas líneas vean las posibilidades del mercado. Como punto de partida por defecto les propongo que visiten la página *Documentación de las AMI de aprendizaje profundo de AWS*⁶³ donde Amazon mantiene la última versión de manual en PDF y HTML para ayudarnos a ejecutar un Jupyter Notebook en su *Cloud* usando una instancia *p2.xlarge* que permite acceso a GPUs por un coste económico razonable, pero suficiente, para probar los *notebooks* de este libro.

Pero nuevamente, quiero recordar que si el lector no dispone de GPU, adjunto

⁶³ Documentación AMI de aprendizaje profundo en AWS. [online]. Disponible en: https://aws.amazon.com/es/documentation/dlami/?nc1=h_ls [Accedido: 1/04/2018]

al código del libro proporcionamos para las redes neuronales que requieren muchos recursos de computación, los ficheros que contienen los pesos de las redes entrenadas y que el lector podrá descargar y usar.

Y ahora que tenemos el entorno listo, solo me queda decirles *Enjoy!*

