Actas de las Jenui, vol. 6. 2021.
Páginas: 227-234

Proyectos de aprendizaje profundo usando datos regionales

Jónathan Heras
Departamento de Matemáticas y Computación
Universidad de La Rioja
26006 Logroño
jonathan.heras@unirioja.es

Resumen

Debido al impacto de las técnicas de aprendizaje profundo, tanto en entornos industriales como académicos, hay una gran demanda de graduados con habilidades en este campo de la inteligencia artificial. Es por esto que las universidades están comenzando a ofertar asignaturas que incluyen temas relacionados con el aprendizaje profundo. En estas asignaturas, las prácticas son fundamentales; sin embargo, la mayoría de dichas prácticas tienen dos inconvenientes. El primero es el uso de, o bien, "datos de juguete" que sirven para enseñar conceptos pero cuyas soluciones no generalizan a problemas reales; o bien, datos que requieren un conocimiento experto para comprender correctamente su contexto. En segundo lugar, la mayoría de prácticas de aprendizaje profundo se centran en la tarea de entrenar un modelo, y no tienen en cuenta otras tareas, como son la limpieza de los datos o el despliegue de los modelos. En este trabajo presentamos una experiencia en una asignatura de inteligencia artificial donde hemos abordado los problemas anteriores usando datos del gobierno de la comunidad autónoma donde se encuentra nuestra universidad. En concreto, los estudiantes han llevado a cabo diversos proyectos de visión por computador y procesado de lenguaje natural usando técnicas de aprendizaje profundo; por ejemplo, han creado un clasificador de noticias o una aplicación para colorear imágenes antiguas. Compartimos aquí el flujo de trabajo utilizado para organizar la experiencia, las lecciones aprendidas y los retos que pueden encontrarse al intentar llevar a cabo iniciativas similares.

Abstract

Due to the impact of Deep Learning both in industry and academia, there is a growing demand of graduates with skills in this field, and Universities are starting to offer courses that include Deep Learning subjects. Hands-on assignments that teach students how to tackle Deep Learning tasks are an instrumental part of those courses. However, most Deep Learning assignments have two main drawbacks. First, they use either toy datasets that are useful to teach concepts but whose solutions do not generalise to real problems, or employ datasets that require specialised knowledge to fully understand the problem. Secondly, most Deep Learning assignments are focused on training a model, and do not take into account other stages of the Deep Learning pipeline, such as data cleaning or model deployment. In this work, we present an experience in an Artificial Intelligence course where we have tackled the aforementioned drawbacks by using datasets from the regional council where our University is located. Namely, the students of the course have developed several computer vision and natural language processing projects; for instance, a news classifier or an application to colourise historical images. We share the workflow followed to organise this experience, several lessons that we have learned, and challenges that can be faced by other instructors that try to conduct a similar initiative.

Palabras clave

Datos regionales, aprendizaje profundo, prácticas, experiencia docente

1. Introducción

Las técnicas de aprendizaje profundo, en inglés *deep learning*, se han convertido en el estándar para abordar problemas en distintos contextos como son la visión por computador [19], el procesado de lenguaje natural [12], o la bioinformática [24]. Debido a su éxito, hay una creciente demanda de expertos en este campo del aprendizaje automático, y gran parte de las universidades están incluyendo temas de aprendizaje profundo en asignaturas de grado y máster.

Para el diseño de asignaturas sobre aprendizaje profundo es conveniente mirar el diseño que han seguido asignaturas de campos estrechamente relacionados como es la ciencia de datos. En dicho campo existen diversos estudios que enfatizan la importancia de la enseñanza práctica y basada en aplicaciones [7, 11],

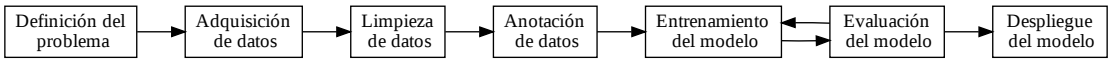

Figura 1: Flujo de trabajo de un proyecto de aprendizaje profundo supervisado.

y esto puede extrapolarse a las asignaturas de aprendizaje profundo. Esto se consigue, en la mayoría de asignaturas, mediante prácticas o proyectos donde los estudiantes tienen que entrenar modelos de aprendizaje profundo usando datos obtenidos de plataformas como *Kaggle*, *Amazon Datasets* o *Google Dataset Search*. Sin embargo la construcción de un modelo es solo uno de los pasos necesarios para abordar un proyecto de aprendizaje profundo (en la figura 1 se resume el flujo de trabajo de un proyecto de aprendizaje profundo), y el resto de pasos no son tenidos en cuenta en la mayoría de prácticas y proyectos. Por ejemplo, las tareas de adquisición, limpieza y etiquetado de datos no son llevadas a cabo por los estudiantes debido a que los datasets disponibles en las diversas plataformas se encuentran ya preprocesados y listos para usar. Otro ejemplo de tarea que no se lleva a cabo es el despliegue de los modelos ya que estos solo son vistos y evaluados por el profesorado, y por lo tanto no se usan fuera del contexto de la asignatura. Un problema adicional es que la mayor parte de datasets públicos son, o bien, "'problemas de juguete" que tienen un interés limitado para los estudiantes, o requieren un conocimiento sobre el campo donde los datos fueron adquiridos para poder sacarles partido. Estos problemas pueden ser abordados usando datos locales o regionales.

Muchas ciudades y regiones en todo el mundo están invirtiendo una gran cantidad de recursos en publicar libremente sus datos, en proyectos de *ciudades inteligentes* [5]. Esto abre la puerta a aplicar distintas técnicas para resolver cuestiones de interés para los ciudadanos, administraciones, negocios e investigadores. El uso en clase de datos urbanos o regionales para construir modelos se aproxima bastante a un proyecto real ya que los datos deben ser limpiados; y para ser útil a la sociedad, los modelos creados deben ser fáciles de usar. Además, estos datos tienen la ventaja adicional de tratar cuestiones familiares para los estudiantes.

En este artículo presentamos una experiencia llevada a cabo en una asignatura de inteligencia artificial donde los estudiantes han desarrollado, de principio a fin y usando técnicas de aprendizaje profundo, diversos proyectos de visión por computador y procesado de lenguaje natural usando datos de la comunidad autónoma donde se encuentra nuestra universidad. Además presentamos las herramientas y el flujo de trabajo empleados para organizar la experiencia. Finalmente, terminamos presentando los retos a los que nos hemos enfrentado y las lecciones aprendidas.

2. Trabajo relacionado

En la actualidad existe una demanda creciente tanto en la industria como en el entorno académico de especialistas en aprendizaje profundo, así como de áreas relacionadas como son el aprendizaje automático, la inteligencia artificial y la ciencia de datos [22]. Esto ha llevado al desarrollo de cursos abiertos en plataformas como Coursera o edX [22], la creación de grados y másteres, y la incorporación de temas relacionados con el aprendizaje profundo al currículo de diversas asignaturas [4].

Un reto que aparece en el diseño de una asignatura que aborde temas de aprendizaje profundo es la creación de prácticas que no estén basada únicamente en ejemplos de juguete. Las tareas basadas en estos ejemplos sirven para ilustrar y enseñar conceptos; sin embargo, las soluciones que se dan para esos ejemplos normalmente no generalizan a situaciones más complejas. Por esto es necesario diseñar tareas que muestren como las técnicas aprendidas en los ejemplos de juguete se pueden extrapolar a problemas reales. Para abordar este reto podemos usar ideas empleadas en asignaturas de ciencia de datos donde existen proyectos para enseñar estadística [16], visualización de datos [9], o búsqueda de información [6].

Una cuestión importante a la que nos enfrentamos a la hora de crear dichas tareas es la búsqueda de bancos de datos, o datasets, que resulten interesantes. Existen diversas fuentes y aproximaciones para realizar dicha búsqueda: plataformas de datasets públicos (por ejemplo, Kaggle o Google Dataset Search), crear un dataset propio [18], o usar datos abiertos [15, 20]. Las tres alternativas tienen sus ventajas e inconvenientes. Los datasets públicos disponibles en plataformas como Kaggle son interesantes, pero los proyectos creados con dichos datasets se centran en la tarea de entrenar un modelo, y requieren conocimientos previos sobre el contexto de los datos para poderles sacar su máximo partido. La segunda alternativa, que consiste en que los estudiantes creen sus propios datasets, consigue que los estudiantes se involucren en el proyecto ya que este surge de sus propias ideas. Sin embargo,

el proceso de generar datos suficientes es una tarea que requiere bastante tiempo y recursos. En este trabajo exploramos la última alternativa: el uso de datos abiertos.

Los datos abiertos regionales son familiares para los estudiantes, y pueden servir para mostrar todas las fases de un proyecto de aprendizaje automático [20]. Sin embargo puede resultar complejo encontrar los datos suficientes para llevar a cabo proyectos interesantes [15]. Además la mayoría de datos abiertos que son liberados son datos estructurados, y los algoritmos de aprendizaje profundo no son especialmente útiles con este tipo de datos sino que brillan cuando se aplican a datos no estructurados como imágenes o texto. En este artículo, presentamos nuestra experiencia donde abordamos los problemas anteriores involucrando a personal de la comunidad autónoma. En concreto, dicho personal nos ayudó a definir los problemas a analizar y a identificar las fuentes de datos a utilizar.

3. Descripción de la experiencia

Esta iniciativa fue llevada a cabo en el contexto de la asignatura de Inteligencia Artificial, optativa del grado en Ingeniería Informática y del grado en Matemáticas. En esta asignatura había cuatro temas: búsqueda en espacio de estados, aprendizaje automático, visión por computador, y procesado del lenguaje natural. En los dos últimos temas se presentan tanto las técnicas tradicionales como las más modernas basadas en aprendizaje profundo. El objetivo de la experiencia fue el desarrollo completo de proyectos de visión por computador y procesado de lenguaje natural empleando técnicas de aprendizaje profundo. Cabe destacar que las técnicas de aprendizaje profundo explicadas en el curso se centraban en los problemas de clasificación de imágenes y texto; sin embargo, los proyectos llevados a cabo por los estudiantes requerían la aplicación de otro tipo de técnicas de aprendizaje profundo.

En la asignatura estaban matriculados 25 estudiantes. Los estudiantes se dividieron en 9 grupos de entre 2 y 4 personas para realizar proyectos sugeridos por el personal de la Comunidad Autónoma donde se encuentra nuestra universidad. Un resumen de dichos proyectos puede verse en el cuadro 1. Los 9 proyectos se pueden clasificar en 5 tareas: reconocimiento facial y búsqueda de información (1 proyecto), coloreado de imágenes (2 proyectos), segmentación de imágenes (1 proyecto), clasificación de imágenes (4 proyectos), y clasificación de texto (1 proyecto).

3.1. Descripción de los proyectos

Proporcionamos a continuación una breve descripción de los proyectos llevados a cabo y de las técnicas empleadas por los estudiantes para abordarlos.

Búsqueda de personas en imágenes. Dada la foto de una persona, el objetivo de este proyecto era identificar todas las imágenes del portal de noticias de la comunidad autónoma donde aparecía dicha persona. Este proyecto requería la aplicación de técnicas de detección y reconocimiento facial, como OpenFace [8] o DeepFace [23], y también métodos del campo de la búsqueda de imágenes; por ejemplo, *VP-trees* [2].

Coloreado de imágenes. En la experiencia ha habido dos proyectos relacionados con el coloreado de imágenes antiguas. El primero se centraba en imágenes aéreas disponibles gracias a la infraestructura de datos espaciales de la comunidad autónoma; y el segundo se centró en imágenes del registro de fotografías de la comunidad (una página web gestionada por la comunidad donde cualquier puede subir imágenes antiguas de la región). Estos proyectos se llevaron a cabo utilizando ideas del proyecto DeOldify[1].

Detección de casas en imágenes aéreas. Los integrantes de este proyecto se enfrentaban al problema de detectar casas en imágenes aéreas proporcionadas por la infraestructura de datos espaciales de la comunidad. Este proyecto se enmarca en la tarea de segmentación semántica donde la arquitectura U-net es ampliamente utilizada para segmentar edificios en imágenes aéreas [1].

Clasificador de noticias. El equipo encargado de este proyecto debía crear un clasificador de noticias de la página oficial de la comunidad. Hasta ahora, las noticias se organizaban de forma manual en departamentos, y el objetivo del proyecto era automatizar esta tarea. Para ello los estudiantes utilizaron una aproximación llamada ULMFit [13].

Clasificación de imágenes. En la experiencia hubo 4 proyectos relacionados con la tarea de clasificar imágenes. Dos de los proyectos usaron imágenes del registro fotográfico de la comunidad; uno para clasificar las imágenes en un conjunto de categorías fijo (incluyendo categorías como colegio, familia, o militar); y el otro para datar las imágenes. Los otros dos proyectos emplearon imágenes de obras del museo, y se encargaban de clasificar las imágenes dependiendo del tipo de pieza (escultura, pintura, etc.), y de datar las imágenes en distintas épocas. Los modelos entrenados para estos proyectos utilizaron la aproximación presentada en la librería FastAI[2].

El flujo de trabajo seguido para llevar a cabo estos proyectos se resume a continuación.

3.2. Flujo de trabajo de los proyectos

Como primer paso en esta iniciativa, el profesorado de la asignatura se reunió con el personal de la comu-

[1] https://github.com/jantic/DeOldify
[2] https://www.fast.ai/

Nombre proyecto	♯ miembros	Tareas	Librerías	Despliegue
Búsqueda de personas en imágenes	4	Reconocimiento facial	Keras	Aplicación escritorio
Coloreado de imágenes antiguas	3	Coloreado de imágenes	FastAI	Colab
Coloreado de imágenes aéreas antiguas	3	Coloreado de imágenes	FastAI	Colab
Detección de casas en imágenes aéreas	3	Segmentación de imágenes	FastAI	Colab
Clasificador de noticias	3	Clasificación texto	FastAI	Colab
Clasificación de imágenes antiguas	2	Clasificación de imágenes	FastAI	Colab
Datando imágenes antiguas	2	Clasificación de imágenes	FastAI	Binder
Datando obras del museo	2	Clasificación de imágenes	FastAI	Web app
Clasificación de obras del museo	3	Clasificación de imágenes	FastAI	Colab

Cuadro 1: Lista de proyectos llevados a cabo en la experiencia

nidad autónoma para identificar un conjunto de tareas, y sus datasets asociados. Estas reuniones fueron fundamentales para encontrar un conjunto de problemas que fueran abordables en la asignatura. En concreto, nos centramos en proyectos en los que existieran datos suficientes y que estuvieran parcialmente anotados. El siguiente paso consistió en proporcionar a los estudiantes un resumen de los distintos proyectos para que formaran equipos y eligieran un proyecto (los proyectos fueron asignados mediante una estrategia *first-in-first-out*). Después de la asignación, el profesorado creo una serie de tareas a través de GitHub classroom[3] que da acceso a un repositorio privado de GitHub a cada equipo. Cada repositorio solo contenía un fichero LÉEME que explicaba cómo acceder a los datos del proyecto, y algunas ideas y enlaces donde se explicaba cómo abordar el proyecto.

La primera tarea a la que se enfrentaron los estudiantes fue el proceso de adquirir, limpiar y anotar sus datos. Todos los datasets estaban disponibles a través de APIs públicas; sin embargo, la anotación asociada no estaba accesible desde el mismo sitio. Por ejemplo, en los proyectos del museo, existe una base de datos para cada obra del museo donde se proporciona su nombre y tipo, y un enlace a una imagen de la obra; sin embargo, las imágenes de las obras están disponibles en un repositorio distinto, por lo que los estudiantes tuvieron que emparejar ambas fuentes de información. Además la mayoría de los datasets se encuentra parcialmente anotados por lo que los estudiantes tuvieron que hacer una limpieza previa antes de utilizarlos.

Una vez que estos pasos de pre-procesado fueron llevados a cabo, los estudiantes debieron entrenar y evaluar sus modelos usando Python como lenguaje de programación, y librerías de aprendizaje profundo como Keras[4] o FastAI. Como entorno de desarrollo, se usaron cuadernos de Jupyter [17], una herramienta *open-source* que se ejecuta en cualquier navegador y que permite combinar bloques de texto y código. Es posible ejecutar los cuadernos de Jupyter de forma local; sin embargo, a la hora de entrenar modelos de aprendizaje profundo es necesario el uso de hardware específico como GPUs o TPUs, y la mayoría de estudiantes no tienen acceso a dichos recursos. Por ello, los cuadernos de Jupyter se ejecutaban de forma online en Google Colaboratory[5], un entorno pre-configurado con las librerías esenciales de aprendizaje automático y aprendizaje profundo que da acceso gratuito a GPUs y TPus a través de una cuenta de Google. Es importante notar que Google Colaboratory se puede conectar con los repositorios de GitHub, por lo que los estudiantes podían guardar fácilmente su progreso en el repositorio de su equipo. Todas las herramientas mencionadas anteriormente habían sido previamente introducidas a los estudiantes en la asignatura. La única funcionalidad desconocida para ellos era el acceso de equipos a un repositorio de GitHub ya que el resto de prácticas del curso eran individuales.

Una vez entrenados los modelos, los estudiantes los evaluaron usando conjuntos de test independientes a los usados para entrenar sus modelos. Todos los modelos salvo el encargado de colorear imágenes antiguas obtuvo una precisión superior al 90 % en su tarea correspondiente (notar que dependiendo del problema se usaban distintas métricas). Por lo tanto, aunque hay espacio para la mejora, se puede considerar que los modelos fueron exitosos y resolvían la tarea asignada.

Finalmente los estudiantes tenían que desplegar sus modelos de manera que fueran fáciles de usar. La mayoría de los equipos, 6 de 9, decidieron usar los formularios de Google Colaboratory, uno creó una aplicación de escritorio, otro una aplicación web, y el último desplegó su modelo usando Binder[6].

Para poder evaluar los proyectos, los estudiantes tuvieron que documentar todo el proceso seguido durante el proyecto en cuadernos de Jupyter que debían ser almacenados en su repositorio de GitHub. Además los estudiantes tenían que presentar su trabajo en una exposición pública, y preparar un vídeo de 2 minutos donde explicaran de manera no técnica su trabajo.

[3] http://classroom.github.com
[4] https://keras.io/
[5] https://colab.research.google.com/
[6] https://mybinder.org/

4. Discusión

En esta sección presentamos las lecciones aprendidas durante este experiencia, y comentamos los retos a los que nos hemos enfrentado.

4.1. Lecciones aprendidas

En primer lugar presentamos las lecciones aprendidas. Estas recomendaciones no se basan únicamente en la opinión del profesorado, sino que también se ha tenido en cuenta el grado de satisfacción de los estudiantes. Para obtener la opinión de los estudiantes se llevó a cabo una encuesta anónima y voluntaria desarrollada con Google Forms. La encuesta consistía de 4 secciones (valoración de GitHub, valoración de los cuadernos de Jupyter y Colab, valoración de los proyectos, y comentarios). Las secciones de valoración consistían de una lista de preguntas que seguían una escala de 4 puntos de Likert yendo de 1 (completamente en desacuerdo) a 4 (completamente de acuerdo), y la sección de comentarios permitía a los estudiantes introducir comentarios adicionales sobre la experiencia. A la encuesta respondieron 15 de los 25 estudiantes.

Pasamos a describir en primer lugar las lecciones aprendidas desde un punto de vista organizativo.

Como sugerían trabajos previos [18], es muy beneficioso involucrar a los estudiantes en proyectos que sean cercanos a su día a día, por lo que el uso de datos regionales es perfecto para esto. Esto se puede ver en las presentaciones de los vídeos donde, por ejemplo, los miembros del equipo encargado de colorear imágenes aéreas mostraron imágenes coloreadas de los pueblos donde viven sus familias; o en el proyecto de búsqueda de personas, donde los miembros del equipo se buscaron a sí mismos en el portal de noticias de la comunidad. En la encuesta anónima, todos los estudiantes afirmaron que habían disfrutado trabajando con datos reales. De hecho, uno de los estudiantes escribió el siguiente comentario "*Los trabajos para mi están genial. Te ayudan a aplicar lo que has aprendido durante el curso a un problema real*".

Una de las mejores decisiones que tomamos fue la de reunirnos con el personal de la comunidad autónoma. Dichas reuniones nos ayudaron a delimitar un conjunto de problemas que encajara con los temas explicados en la asignatura, resolviendo así el problema de encontrar datos suficientes e interesantes en datos abiertos [15]. En estas reuniones, el profesorado pudo explicar al personal de la comunidad lo que los estudiantes podían llegar a hacer, y también establecer ciertos límites a sus expectativas (todos los proyectos eran pruebas de concepto que debían ser desarrolladas en un tiempo limitado, aproximadamente unas 25 horas). En este primer año de la experiencia, solo el profesorado se reunió con el personal de la comunidad. En los próximos años planeamos incorporar a los estudiantes a dichas reuniones ya que ahora el personal de la comunidad sabe qué tipo de proyectos se pueden llevar a cabo. Sin embargo, es necesario que el profesorado siga involucrado en las reuniones para estimar el tiempo necesario para abordar los distintos proyectos debido a que los estudiantes no cuentan con la experiencia necesaria para realizar esas estimaciones.

Otra de las decisiones que resultó ser acertada fue incluir un fichero LÉEME en los repositorios de los estudiantes. Inicialmente cada equipo tenía acceso a un repositorio privado de GitHub que contenía únicamente un fichero LÉEME. En dicho fichero, el profesorado explicaba cómo acceder a los datos, y proporcionaba enlaces e información de cómo abordar el proyecto. Entre la documentación que se proporcionaba a los estudiantes, se incluyeron enlaces a repositorios libres que incluían cuadernos de Jupyter abordando proyectos similares. Nuestra primera recomendación para los estudiantes fue que ejecutaran dichos cuadernos y que los usaran como base para sus proyectos. De este modo, los equipos tenían un punto de partida para saber cómo organizar sus datasets, y entrenar sus modelos. Además el fichero LÉEME contenía enlaces a tareas de ampliación, como el uso de arquitecturas novedosas para la clasificación de imágenes, o la aplicación de métodos de *ensemble*. Estos temas de investigación permitían a los equipos profundizar en las técnicas de aprendizaje profundo.

La última lección aprendida desde el punto de vista organizativo tiene que ver con los vídeos realizados por el estudiantado. Como se ha comentado anteriormente se pidió a los estudiantes que prepararan vídeos cortos y accesibles, desde un punto de vista técnico, donde explicaran su trabajo. El objetivo era doble. Por un lado queríamos que los estudiantes hicieran el esfuerzo de resumir su trabajo de una manera atrayente. Por otro lado, dichos vídeos servían para presentar el trabajo realizado al personal de la comunidad, y mostrar a otros miembros de la comunidad el tipo de tareas que podían ser resueltas mediante técnicas de aprendizaje profundo. Estos vídeos son importantes de cara a continuar con la experiencia en próximos años con nuevos proyectos.

Pasamos ahora a centrarnos en las lecciones aprendidas desde un punto de vista técnico. Estas lecciones pueden aplicarse no solo a proyectos basados en datos abiertos sino a cualquier proyecto de aprendizaje profundo.

El entrenamiento de modelos de aprendizaje profundo desde cero es una tarea que requiere una cantidad ingente de datos y recursos computaciones; y, en la mayoría de los casos, no es posible tener acceso a tal cantidad de recursos. Este problema puede resolverse usando una técnica llamada *transfer learning* [3]. Este

método reusa un modelo entrenado en una tarea origen en una nueva tarea destino. Esto reduce considerablemente la cantidad de datos y tiempo que son necesarios para construir un modelo de aprendizaje profundo preciso. Esta técnica puede ser empleada en prácticamente cualquier proyecto de aprendizaje profundo, por lo que es importante introducirla a lo largo de la asignatura de cara a que los estudiantes puedan emplearla en sus proyectos.

La siguiente lección aprendida tiene que ver con el almacenamiento del código de los estudiantes. En esta experiencia, todo el código y modelos producidos en los proyectos se almacenaron en repositorios de GitHub. En concreto, todos los repositorios pertenecen a una "*organization*" gestionada por el profesorado de la asignatura. De esta manera, los estudiantes solo tienen acceso al repositorio de su equipo, y el profesorado puede acceder a todos los repositorios de los estudiantes. Esta aproximación no solo simplifica la colaboración entre los miembros de un equipo, sino que también mejora la comunicación con el profesorado, ya que pueden acceder directamente al código de los estudiantes cuando estos necesitan ayuda. Además, el profesorado puede seguir el progreso de los estudiantes dado que es posible monitorizar los *commits* realizados y sus autores.

En la encuesta llevada a cabo, el uso de GitHub tuvo una valoración positiva. En concreto, todos los estudiantes estaban satisfechos con el uso de esta herramienta en los proyectos, y pensaban que los repositorios les ayudaron a gestionar mejor su código. Además, la mayoría de los estudiantes (el 93,3 %) consideraban que GitHub es fácil de usar y que facilita el trabajo en equipo. Por último, todos los estudiantes opinaban que esta herramienta debería ser usada en otras asignaturas.

La siguiente lección que hemos aprendido tiene que ver con Google Colaboratory. Este entorno, que proporciona acceso gratuito a GPUs, ha demostrado ser suficiente para entrenar los modelos de aprendizaje profundo desarrollados en los proyectos. Un problema que surge al usar Google Colaboratory es la limitación de 12 horas de uso (es decir, tras 12 horas de usar el entorno, este se reinicia). Esto forzaba a que los estudiantes tuvieran que guardar *checkpoints* intermedios para evitar perder su trabajo ya que en ocasiones 12 horas no eran suficientes para completar el proceso de entrenamiento. Sería posible entrenar modelos por más tiempo, sin necesidad de los puntos de guardado intermedios, usando entornos cloud como Amazon AWS o la plataforma en la nube de Google Cloud (ambos ofrecen créditos gratuitos para su uso por parte de estudiantes); sin embargo, la configuración y gestión de estos entornos es más compleja, requieren información financiera (aunque no se realicen cargos), y se encuentran fuera del alcance de los objetivos de este curso.

También pedimos la opinión de los estudiantes sobre este entorno, y todos los estudiantes encuestados mostraron su satisfacción con el uso de Google Colaboratory. También opinaban que la integración de esta herramienta con GitHub era correcta y fácil de usar.

Sin embargo, este entorno tiene ciertas problemáticas. El aprendizaje profundo está en constante evolución, lo que supone que las librerías que implementan técnicas de aprendizaje profundo se actualicen de forma habitual. Esto tiene grandes beneficios (ya que los errores se solucionan de manera rápida, y se incluyen nuevos métodos que van surgiendo), pero también desventajas (en algunos casos, el código que funcionaba para una versión de una librería deja de funcionar al actualizarla). Estos problemas pueden resolverse en un ordenador local mediante el uso de entornos virtuales donde las versiones de la librería son fijadas en la máquina virtual; pero, Google Colaboratory actualiza de forma habitual las versiones de las librerías incluidas en su máquina virtual, y esto es un problema por dos razones. En primer lugar, algunos estudiantes se encontraban con que parte de su código dejaba de funcionar de un día para otro; y, en segundo lugar, el código proporcionado en los tutoriales online sugeridos por el profesorado, y que eran usados como base por los estudiantes, producían errores al ejecutarse. Para abordar este problema, se proporcionó a los estudiantes un fichero de requisitos que permite instalar versiones concretas de las librerías necesarias en Google Colaboratory por medio del entorno pip[7]. Esto permite a los estudiantes trabajar en un entorno donde es seguro que van a poder reproducir sus resultados.

Otra lección aprendida tiene que ver con las librerías de aprendizaje profundo. En la actualidad existen dos librerías principales de aprendizaje profundo que son usadas tanto en el entorno industrial como académico: Tensorflow y Pytorch [10]. Sin embargo, para proyectos como los presentados en nuestra experiencia, es preferible usar una librería que proporcione una manera sencilla de acceder a distintos tipos de modelos, y que además implemente una serie de buenas prácticas. Durante la asignatura, se presentaron dos de esas herramientas a los estudiantes: Keras y FastAI. Para los proyectos, los equipos podían emplear cualquier librería, pero es destacable que solo uno de ellos usó una librería distinta a FastAI. Esto se debe principalmente a que esta librería proporciona una API amigable y permite entrenar modelos actuales con unas pocas líneas de código.

La última lección aprendida tiene que ver con cómo gestionaron los datos los equipos. En el entorno de Google Colaboratory, los datos solo se conservan en la sesión, por lo los estudiantes deben cargar los datos cada vez que se conectan al entorno. Para abordar este

[7] `https://pypi.org/project/pip/`

problema, les pedimos a los estudiantes que crearan un cuaderno de Jupyter dedicado únicamente a descargar y limpiar los datos, y una vez que estos estaban procesados, los estudiantes debían subir los datos a un sistema online de gestión de ficheros (la mayoría de los equipos usaron OneDrive ya que nuestra universidad ofrece un plan gratuito para los estudiantes, pero otros equipos usaron Google Drive o Dropbox). De esta forma, los miembros de cada equipo, y también el profesorado, podía acceder fácilmente a una versión limpia de los datos.

4.2. Retos

Durante esta iniciativa también nos enfrentamos a una serie de retos que deben ser tenidos en cuenta al realizar este tipo de experiencias.

Uno de los principales retos de esta iniciativa fue la carga de trabajo del profesorado tanto en términos de organización como de supervisión de los proyectos. Por lo tanto, es necesario que el profesorado tenga tiempo para organizar los proyectos (es decir, contactar y reunirse con el personal de la comunidad autónoma, y preparar material para guiar a los estudiantes) y también proporcionar consejo individualizado a cada equipo. Además, en este tipo de proyectos es difícil prever las dificultades que los estudiantes se pueden encontrar, y esto supone una carga adicional de trabajo en términos de supervisión.

Un reto relacionado con la supervisión de los estudiantes es reproducir los errores que los estudiantes se encuentran durante el proceso de entrenamiento. Una de las grandes ventajas de usar los repositorios de GitHub y Google Colaboratory es la posibilidad de acceder de manera sencilla al código de los estudiantes, y, por lo tanto, el profesorado puede revisar de manera sencilla los mensajes de error. Sin embargo, en muchas ocasiones, revisar los mensajes de error no proporciona información suficiente para descubrir el problema, y por lo tanto es necesario ejecutar de nuevo el código de los estudiantes. Esto puede ser un reto ya que en muchos casos es necesario ejecutar una cantidad de código considerable antes de llegar al punto donde los estudiantes habían encontrado un error; además, el estado interno de los cuadernos de Jupyter puede dificultar la tarea de reproducir las condiciones exactas donde los estudiantes encontraron el problema [14]. Para minimizar este reto, nuestra recomendación a los estudiantes fue crear cuadernos específicos para cada uno de sus experimentos. De este modo, era más sencillo ayudarles a abordar los problemas que encontraban.

El último reto tiene que ver con que el estudiantado tuvo que abordar técnicas fuera del currículum. De los 9 proyectos, 4 de ellos requerían que los estudiantes aplicaran métodos de aprendizaje profundo que no habían sido explicados previamente en la asignatura, y los otros 5 proyectos incluían tareas adicionales para profundizar en las tareas de clasificación de imagen y texto. Para abordar esta cuestión, proporcionamos a los estudiantes material online con ejemplos y explicaciones de las técnicas a emplear. Sin embargo, un problema que encontramos con esta aproximación fue que los estudiantes se centraban únicamente en entrenar los modelos sin entender lo que hacían internamente. Una solución para abordar este problema se basa en el trabajo del equipo encargado de detectar casas a partir de imágenes aéreas. Este equipo, además de entrenar un modelo de segmentación, creó un cuaderno donde explicaba la arquitectura U-net [21], que es fundamental para problemas de segmentación, y proporcionó un ejemplo de juguete para explicar cómo usarla. En el futuro, planeamos explorar esta aproximación pidiendo a los estudiantes que hagan algo similar.

5. Conclusiones

En este trabajo hemos presentado una iniciativa para aplicar métodos de aprendizaje profundo a la creación de proyectos basados en el uso de datos regionales. En la actualidad, la mayoría de prácticas de aprendizaje profundo emplean o bien datos de juguete, o datos que requieren un conocimiento previo para comprenderlos correctamente. Además dichas prácticas se centran en la tarea de construir modelos, dejando a un lado otras tareas igualmente importantes para un proyecto de aprendizaje profundo. En nuestra experiencia, los estudiantes han trabajado con datos que les resultan familiares, y han estado involucrados en todas las fases del desarrollo de un proyecto de aprendizaje profundo.

El objetivo de este trabajo era presentar las lecciones aprendidas y los retos que el profesorado se puede encontrar al organizar iniciativas similares que empleen datos abiertos. Entre las lecciones aprendidas cabe destacar el beneficio de involucrar a personal de la comunidad autónoma para elegir un conjunto de problemas que motiven a los estudiantes. En nuestro caso la experiencia será continuada en el futuro con nuevos proyectos y métodos de aprendizaje profundo.

Referencias

[1] Abolfazl Abdollahi, Biswajeet Pradhan y Abdullah M. Alamri: *An ensemble architecture of deep convolutional Segnet and Unet networks for building semantic segmentation from high-resolution aerial images*. Geocarto International, páginas 1–16, 2020.

[2] Adrian RoseBrock: *Building an Image Hashing Search Engine with VP-Trees and OpenCV*, 2019.

[3] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan y Stefan Carlsson: *CNN features off-the-shelf: An astounding baseline for recognition*. En *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, CVPRW'14, páginas 512–519, 2014.

[4] Amy K. Hoover, Adam Spryszynski y Michael Halper: *Deep Learning in the IT Curriculum*. En *Proceedings of the 20th Annual SIG Conference on Information Technology Education*, páginas 49–54, 2019.

[5] Bhagya Nathali Silva, Murad Khan y Kijun Han: *Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities*. Sustainable Cities and Society, 38:697–713, 2018.

[6] Bhavya, Assma Boughoula, Aaron Green y ChengXiang Zhai: *Collective Development of Large Scale Data Science Products via Modularized Assignments: An Experience Report*. En *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, páginas 1200–1206, 2020.

[7] Bina Ramamurthy: *A practical and sustainable model for learning and teaching data science*. En *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, páginas 169–174, 2016.

[8] Brandon Amos, Bartosz Ludwiczuk y Mahadev Satyanarayanan: *OpenFace: A general-purpose face recognition library with mobile applications*. Informe técnico, CMU-CS-16-118, CMU School of Computer Science, 2016.

[9] Deborah Nolan y Jamis Perrett: *Teaching and Learning Data Visualization: Ideas and Assignments*. The American Statistician, 70(3):260–269, 2016.

[10] Horace He: *The State of Machine Learning Frameworks in 2019*. The Gradient, 2019.

[11] Il-Yeol Song y Yongjun Zhu: *Big data and data science: what should we teach?* Expert Systems, 33(4):364–373, 2016.

[12] Jacob Devlin, Ming Wei Chang, Kenton Lee y Kristina Toutanova: *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. En *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volumen 1, páginas 4171–4186, 2019.

[13] Jeremy Howard y Sebastian Ruder: *Universal Language Model Fine-tuning for Text Classification*. En *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, volumen 1, páginas 328–339, 2018.

[14] Joel Grus: *I don't like notebooks*. En *The official Jupyter conference*, 2018.

[15] Joelle Pineau y Pierre Luc Bacon: *Analyzing Open Data from the City of Montreal*. En *Proceedings of the 2nd International Workshop on Mining Urban Data*, páginas 11–16, 2015.

[16] Johanna Hardin, Roger Hoerl, Nicholas J. Horton y Deborah Nolan: *Data science in statistics curricula: Preparing students to think with data*. The American Statistician, 69(4):343–353, 2015.

[17] Jupyter Development Team: *Jupyter Notebooks — a publishing format for reproducible computational workflows*. En *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, páginas 87–90, 2016.

[18] Kaiman Zeng, Yancheng Li, Yida Xu y Nansong Wu Di Wu: *Introducing AI to Undergraduate Students via Computer Vision Projects*. En *Proceedings of the Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, páginas 7956–7957, 2018.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren y Jian Sun: *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*. En *Proceedings of the IEEE International Conference on Computer Vision*, páginas 1026–1034, 2015.

[20] Matthew Love, Charles Boisvert, Elizabeth Uruchurtu y Ian Ibbotson: *Nifty with Data: Can a Business Intelligence Analysis Sourced from Open Data Form a Nifty Assignment?* En *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*, páginas 344–349, 2016.

[21] Olaf Ronneberger, Philipp Fischer y Thomas Brox: *U-Net: Convolutional Networks for Biomedical Image Segmentation*. En *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, volumen 9351 de *Lecture Notes in Computer Science*, páginas 234–241, 2015.

[22] Roberta Kwok: *Junior AI Researchers are in Demand by Universities and Industry*. Nature, 568(7753):581–583, 2019.

[23] Yaniv Taigman, Ming Yang, Marc Ranzato y Lior Wolf: *DeepFace: Closing the Gap to Human-Level Performance in Face Verification*. En *2014 IEEE Conference on Computer Vision and Pattern Recognition*, páginas 1701–1708, 2014.

[24] Yu Li, Chao Huang, Lizhong Ding, Zhongxiao Li, Yijie Pan y Xin Gao: *Deep learning in bioinformatics: Introduction, application, and perspective in the big data era*. Methods, 166:4–21, 2019.