

MÓDULO DE DEEP LEARNING: RETOS PARA EL ESTUDIANTE

MANUEL UJALDÓN

1.	Clasificación de imágenes	2
1.1.	Introducción	2
1.2.	Herramientas y desarrollo	2
1.3.	Sugerencias	3
2.	Detección de objetos	3
2.1.	Introducción	3
2.2.	Herramientas y desarrollo	3
2.3.	Sugerencias	3
3.	Sistemas de recomendación	4
3.1.	Introducción	4
3.2.	Herramientas y desarrollo	4
3.3.	Sugerencias	5

A lo largo de este documento propondremos una serie de retos al estudiante del Módulo de Deep Learning de Nvidia cursado a través del workshop del *Deep Learning Institute (DLI)* titulado *Fundamentals of Deep Learning*. Se persigue con ello que el alumno consolide y amplíe su formación desarrollando una serie de ideas prácticas dentro de la plataforma de *Amazon Web Services (AWS)* que la empresa pone a su disposición dentro de su cuenta del DLI, tanto a nivel software (herramientas y recursos ya instalados) como hardware (GPUs de última generación), por un período de cortesía de entre seis meses y un año a la conclusión del módulo ¹.

Al ofrecerse todos los contenedores, también se puede optar por instalarlo todo en tu máquina local y trabajar sin restricciones temporales.

Se plantean retos estructurados en tres áreas de dificultad creciente:

❶ **Clasificación de imágenes.**

[<https://catalog.ngc.nvidia.com/orgs/nvidia/collections/imageclassification>]

❷ **Detección de objetos.**

[<https://catalog.ngc.nvidia.com/orgs/nvidia/collections/objectdetection>]

❸ **Sistemas de recomendación.**

[<https://catalog.ngc.nvidia.com/orgs/nvidia/collections/recommendersystems>]

Alternativamente, el alumno puede encontrar más de 30 propuestas adicionales dentro de las colecciones del *NGC Catalog* de Nvidia [<https://catalog.ngc.nvidia.com/collections>], y elegir cualquier otra que le resulte de especial interés.

RETO 1

Clasificación de imágenes

1.1 ► Introducción

A lo largo del curso ya hemos hecho algunos proyectos relacionados con esta temática, ilustrando cómo las redes neuronales convolucionales (CNNs) y los modelos entrenados previamente pueden servirnos para mejorar los resultados. El reto ahora consiste en utilizar redes neuronales residuales (*ResNets - Residual Networks*), una variante de las CNNs en la que las conexiones de las neuronas se articulan también entre capas no adyacentes. Estas nuevas conexiones tratan de mover el gradiente de la función de coste de forma más agresiva hacia su mínimo, mejorando tanto la precisión como el tiempo necesario para entrenar el modelo.

Dentro de las redes residuales utilizaremos ResNet-50, una variante dotada de 50 capas.

1.2 ► Herramientas y desarrollo

Los contenedores, modelos y recursos que se ponen a disposición del alumno para llevar a cabo este reto se encuentran disponibles en el enlace Web [<https://catalog.ngc.nvidia.com/orgs/nvidia/collections/imageclassification/entities>].

Por otra parte, el siguiente repositorio de Nvidia contiene todos los elementos necesarios para trabajar con este tipo de modelos, e indica paso a paso cómo ponerlos en juego.

[<https://github.com/NVIDIA/DeepLearningExamples/tree/master/PyTorch/Classification/ConvNets/resnet50v1.5>].

¹ El período exacto de uso gratuito de todos estos recursos ha venido variando entre estas dos cotas (6 y 12 meses) en los últimos años según la política de la empresa y los costes asociados que le supone, sin que podamos precisar cuál de ellas está vigente en cada momento. Pero aún en el peor caso, creemos que seis meses son suficientes para que los alumnos motivados puedan aprovechar toda la infraestructura que Nvidia pone a su disposición

Sugerencias

◀ 1.3

Se puede optar por cualquiera de las dos opciones siguientes, o incluso combinar ambas:

- 1 Utilizar los conjuntos de datos que se encuentran en el repositorio y mejorar u optimizar la evaluación experimental que se propone como ejemplo.
- 2 Cambiar el conjunto de datos para tratar otro problema diferente. Entre los más populares se encuentran los relacionados con el diagnóstico médico (como la monitorización de ciertos rasgos de un paciente a partir de sus muestras), la búsqueda visual de contenido similar al de una imagen dada (para comprar *on-line* productos afines al deseado por el cliente) y la categorización de imágenes desclasificadas en grandes bases de datos.

Detección de objetos

RETO 2

Introducción

◀ 2.1

En la primera versión del curso *Fundamentals of Deep Learning* del DLI, el último proyecto consistía en detectar objetos utilizando el modelo *DetectNet* y una estrategia de ventana deslizando sobre una imagen para localizar el objeto. Resultaba así un proceso bastante lento, con la dificultad añadida de manipular la estructura de datos necesaria para representar el objeto acotado en la imagen.

Dada la importancia de esta tarea, desde entonces se han desarrollado infinidad de mejoras, siendo *SSD* (*Single Shot Detector*) una de las más notables. Este modelo se alimenta hacia adelante para producir un conjunto fijo de cajas acotadas junto a una puntuación de confianza para la ubicación del objeto en su interior, aplicando finalmente un filtro de supresión no máximo para sacar las detecciones finales.

La red SSD consta de dos redes, una apilada encima de la otra. La primera es una CNN que extrae los rasgos de la imagen de forma similar a las redes que clasifican imágenes. La segunda utiliza capas convolucionales cada vez más pequeñas para detectar objetos de diferentes tamaños.

Herramientas y desarrollo

◀ 2.2

Los contenedores, el modelo SSD y el resto de recursos necesarios para realizar esta actividad se suministran en la página Web [<https://catalog.ngc.nvidia.com/orgs/nvidia/collections/objectdetection/entities>] y están disponibles tanto para PyTorch como para TensorFlow, que es el API que hemos utilizado durante la impartición de nuestro módulo y con el que el alumno seguramente se sentirá más cómodo.

Sugerencias

◀ 2.3

Se pueden presentar infinidad de aplicaciones reales en las que se luzca el modelo propuesto. Como ejemplos ilustrativos, Nvidia menciona cuatro áreas de especial relevancia:

- 1 **Conducción autónoma.** Sin duda, una de las aplicaciones en las que más capital humano ha invertido en los últimos años. Las cámaras del vehículo deben detectar otros vehículos, peatones, todo tipo de señales de tráfico y obstáculos en la calzada, y además ubicar en tiempo récord su ubicación exacta. Una aplicación sencilla a realizar aquí es la de entrenar

la red para reconocer el amplio abanico de señales del código de la circulación, primero con fotos de las señales extraídas de libros y luego con fotos obtenidas durante la circulación en tráfico real.

- ② **Clasificación en almacenes.** Numerosos procesos industriales contemplan el envasado, ordenación o inventariado de objetos que pasan a través de una cinta transportadora, y que gracias a la inteligencia artificial puede llevarse a cabo por parte de robots.
- ③ **Video-vigilancia.** La detección de objetos permite analizar de forma automatizada largos vídeos para localizar la presencia de algún elemento de interés o anomalía.
- ④ **Tareas peligrosas.** En la manipulación de objetos y realización de tareas peligrosas en plantas de reciclaje o de energía nuclear, por poner dos buenos ejemplos, los robots también pueden sustituir al humano si tienen la habilidad para detectar los elementos a manipular.

RETO 3

Sistemas de recomendación

3.1 ► Introducción

Los sistemas de recomendación se trataron de forma breve en el último capítulo del curso dedicado a las arquitecturas avanzadas. Tratan de filtrar la información para cuantificar la preferencia que un usuario otorga a cierto elemento, permitiendo personalizar su experiencia. Para ello, debemos enfrentarnos a múltiples desafíos:

- Por el lado de los datos, uno cuantitativo, las enormes bases de datos, y otro cualitativo, su complejo preprocesamiento.
- Por el lado del coste computacional, la carga de datos puede ser lo más lento si no se optimiza bien, y el proceso en su conjunto requiere completar numerosas tareas repetitivas de entrenamiento y evaluación que corren el riesgo de eternizarse.

Entre los modelos más exitosos que se han utilizado para este propósito, cabe mencionar:

- ① **Wide and Deep.** Modelos lineales amplios y redes neuronales entrenados conjuntamente.
- ② **Deep Learning Recommendation Model (DLRM).** El favorito de Nvidia, que forma parte de su entorno de trabajo Merlin [<https://developer.nvidia.com/nvidia-merlin>], para construir sistemas de recomendación de alto rendimiento.
- ③ **Neural Collaborative Filtering (NCF),** un modelo genérico que puede expresar y generalizar la factorización matricial.
- ④ **Variational Autoencoder (VAE),** extendido al filtrado cooperativo para una realimentación implícita.
- ⑤ **BERT4Rec,** una recomendación secuencial con representaciones bidireccionales procedentes de Transformers, cuya arquitectura vimos por encima en el tramo final del curso dedicado a las arquitecturas avanzadas.

3.2 ► Herramientas y desarrollo

Los cinco modelos mencionados anteriormente están disponibles en la infraestructura que Nvidia pone a tu disposición en el enlace Web [<https://catalog.ngc.nvidia.com/orgs/nvidia/collections/recommendersystems/entities>], proporcionando también contenedores para operar, una vez más, tanto en PyTorch como en TensorFlow.

Sugerencias

◀ 3.3

Todos estamos familiarizados con este tipo de aplicaciones cuando compramos por Internet, ya que nos ofrecen una canción acorde a nuestros gustos musicales, una prenda afín a otra que ya hemos comprado, o una película de nuestros actores favoritos. Pero también podemos ir más allá para sugerir experiencias en viajes según el destino y la conducta previa del usuario, o servicios financieros en función del perfil del cliente y su situación económica, por poner un par de ejemplos más del montón que se te pueden ocurrir a tí.