Actividad 1. Laboratorio: *Fine-tuning* de modelos de visión artificial en la nube con AWS SageMaker Studio Lab

**Objetivos**

En esta actividad vas a trabajar con la plataforma de inteligencia artificial en la nube de Amazon Web Services (AWS), SageMaker. En particular, utilizaremos el entorno de experimentación gratuito **SageMaker Studio Lab**, a través del que tienes acceso a infraestructura de computación de Amazon (CPU y GPU) y a una instancia de Jupyter Lab.

Con esta actividad podrás llevar a cabo el proceso de *fine-tuning* de un modelo preentrenado en Pytorch, el **DenseNet-161**. Este modelo se utiliza en visión artificial para clasificación de imágenes, detección de objetos y segmentación semántica. En este ejemplo lo utilizarás para clasificar imágenes desde el punto de vista meteorológico.

De forma general, esta actividad te permite familiarizarte con un entorno de IA basado en la nube, y utilizarlo para tareas que requieren una capacidad de computación superior a la proporcionada por un PC convencional.

**Pautas de elaboración**

La realización de esta actividad comprende las siguientes actividades:

1. Puesta en marcha el entorno de SageMaker Studio Lab.
2. Clonado del repositorio de ejemplos proporcionado por Amazon.
3. Análisis del código fuente del *notebook Train an image classification model with PyTorch.*
4. Ejecución del *notebook.*
5. Cumplimentación de la batería de preguntas incluidas al final de esta sección.

A continuación, se detalla cada uno de estos pasos:

**Registro en SageMaker Studio Lab y Kaggle**

El primer paso es enviar una solicitud para obtener una cuenta gratuita en SageMaker Studio Lab. La respuesta a esta solicitud puede tardar varios días. Las credenciales de Studio Lab llegarán al correo electrónico que hayas especificado en la solicitud.

La solicitud puede enviarse desde la siguiente URL: <https://studiolab.sagemaker.aws/requestAccount>

A continuación, abrir una cuenta gratuita en **Kaggle,** esta se utiliza para descargar el *dataset* que se va a utilizar en esta actividad.

La cuenta puede crearse desde la siguiente URL: <https://www.kaggle.com/account/login?phase=startRegisterTab&returnUrl=%2F>

**Clonado del repositorio de ejemplos**

Para iniciar una sesión de trabajo con SageMaker Studio Lab, hacer clic en «Start runtime» en la página principal, después de hacer «Login».

No olvides finalizar la sesión cuando hayas terminado la actividad. Para ello, puedes hacer clic en «Stop runtime», como se muestra en la Figura 1:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Figura . Página principal de SageMaker Studio Lab. Fuente: SageMaker Studio Lab (s. f.). *Learn and experiment with machine learning.* <https://studiolab.sagemaker.aws/>

El *notebook* con el que vamos a trabajar está preparado para funcionar en SageMaker Studio Lab.

Está alojado en la siguiente URL de GitHub:

<https://github.com/aws/studio-lab-examples>

En la sección Computer Vision, elegir el *notebook* «Weather Classification for Disaster Risk Reduction with DenseNet-161», y hacer clic en «Open Studio Lab».

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Figura . Repositorio GitHub de ejemplos de AWS SageMaker Studio Lab. Fuente: Aws (s. f.). GitHub - aws/studio-lab-examples: Example notebooks for working with SageMaker Studio Lab. GitHub. <https://github.com/aws/studio-lab-examples>

A continuación, selecciona «Copy to project». Se abrirá Jupyter Lab. En el cuadro de diálogo que se abre, debes seleccionar «Clone Entire Repo».

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

Figura . Entorno JupyterLab de AWS SageMaker Studio Lab. Fuente: JupyterLab, s. f.

Navega hasta la siguiente ruta en el explorador de ficheros:

studio-lab-examples/computer-vision/weather-data

Abre el *notebook* titulado weather-image-classification-pytorch.ipynb. El *kernel* que debes utilizar para ejecutar este *notebook* es sagemaker-distribution:Python.

**Análisis del código fuente**

Lee atentamente las instrucciones del *notebook*. En este punto, deberás acceder a Kaggle para descargar el *dataset* que vamos a utilizar en la actividad para realizar el *fine-tuning* del modelo.

Accede a través de la siguiente URL: <https://www.kaggle.com/datasets/pratik2901/multiclass-weather-dataset>

Debes subir el fichero archive.zip a la misma ruta en la que se encuentra el *notebook*.

**Ejecución del *notebook***

Ahora debes ejecutar de forma secuencial cada celda del *notebook*. Es importante que compruebes en cada paso que no se produce ningún error.

La primera celda instala varios paquetes con dependencias. Es posible que tengas que reiniciar el *kernel* después haberla ejecutado. De otro modo, pueden aparecer errores en las celdas sucesivas.

**Lista de preguntas**

Responde a cada pregunta que se formula a continuación. Cuando se pida una explicación, debes proporcionar al menos un párrafo en la respuesta.

1. **Sección 1:** *Install packages.* ¿Cuál es la finalidad de la llamada a torch.cuda.is\_available()? ¿Qué valor retornaría esta llamada si seleccionamos GPU como *Compute type* en SageMaker Studio Lab? ¿Qué es CUDA? ¿Qué ventaja principal ofrece CUDA en el ámbito de IA?
2. **Sección 2.1:** *Reading Image Data.* ¿Cuál es el contenido de la variable images\_population? ¿Qué significan los valores de este contenido? ¿Qué finalidad tiene la variable le?
3. **Sección 2.3:** *Create WeatherDataset class.* ¿Qué relación existe entre las clases WeatherDataset y Dataset (de torch.utils.data)? ¿Qué valor devuelve el método \_\_len\_\_() de WeatherDataset?
4. **Sección 3:** *Dataset Normalization.* ¿Para qué sirve cada llamada a *transforms* en transforms.Compose([transforms.Resize((230,230)), transforms.ToTensor()])?
5. **Sección 3:** *Dataset Normalization.* ¿Por qué se crea un DataLoader a partir del Dataset en torch.utils.data.DataLoader(dataset, batch\_size=batch\_size, num\_workers=4)?
6. **Sección 3:** *Dataset Normalization.* ¿Qué función cumple cada una de las siguientes llamadas? RandomRotation(), RandomResizedCrop(), RandomHorizontalFlip(). ¿Por qué se aplican exclusivamente a los datos de entrenamiento?
7. **Sección 4:** *Load the DenseNet-161*. ¿Por qué fijamos a True el parámetro pretrained en model\_ft = models.densenet161(pretrained=True)?
8. **Sección 4:** *Load the DenseNet-161*. ¿Bajo qué condición se guarda un nuevo *checkpoint* del modelo?
9. **Sección 6:** *Test a model.* ¿Por qué se utiliza en la fase de pruebas la llamada a torch.no\_grad()?
10. **Sección 7:** *Show predictions graphs and confusion matrix*. ¿Qué significan los valores de la última fila de la matriz de confusión?

**Extensión y formato**

El entregable de la actividad debe incluir cada una de las preguntas de la sección anterior con su correspondiente respuesta. Salvo en los casos en los que se solicite un dato puntual, la respuesta debe incluir **un párrafo** con la explicación solicitada. **No se pueden incluir capturas de pantalla en la respuesta.** Adicionalmente, se debe incluir como adjunto un *export* del *notebook* ejecutado, **en formato HTML**.

La extensión **máxima** del documento es de 4 páginas.

Rúbrica

|  |  |  |  |
| --- | --- | --- | --- |
| *Fine-tuning* de modelos de visión artificial en la nube con AWS SageMaker Studio Lab | Descripción | Puntuación máxima  (puntos) | Peso  % |
| Criterio 1 | Entrega del *export* con todas las celdas del *notebook* ejecutadas de forma secuencial y sin errores en la salida. | 2 | 20 % |
| Criterio 2 | Entrega de una respuesta correcta a cada una de las 10 preguntas formuladas. | 8 | 80 % |
|  |  | **10** | **100 %** |