Deep Learning

Pytorch3 - Datasets, Dataloaders, Collate function

UNIVERSIDAD
POLITÉCNICA
DE MADRID

Máster
Deep Learning

Índice

1. Introducción a Datasets y Dataloaders

Introducción a Datasets y Dataloaders

- **Dataset**: clase de pytorch diseñada para representar y manejar conjuntos de datos. Es decir, proporciona un acceso estructurado a los datos, *sin cargarlos directametne en memoria*.

- **Dataloader**: clase de pytorch que *carga eficientemente* los datos desde un Dataset, organizandolo en batches, procesarlos, etc.

¿Porqué es tan importante?

1. Permite manejar grandes volumenes de datos
2. Flexibilidad
3. Eficiencia

2. Datasets en Pytorch

Datasets precargados

- Se tratan de dataset ya disponibles en la librería de Pytorch y no necesitan están guardados en local.
- Tipos: Texto, imagen, audio o video (TorchVision)
- Permiten realizar transformaciones directamente sobre los datos utilizando.
  - Cambiar el tamaño
  - Aumento de datos (data augmentation)
  - Normalizar
  - Etc.

Datasets personalizados

- Nos permite trabajar con datos que tenemos guardados en local

- Necesita crear una Clase de tipo Dataset compuesta de tres funciones:
  - **__init__**: definiremos nuestros datos y en caso de estar en archivo los cargaremos
  - **__len__**: nos proporcionará el número de muestras de
  - **__getitem__**: devolverá muestras del dataset a partir de los índices.

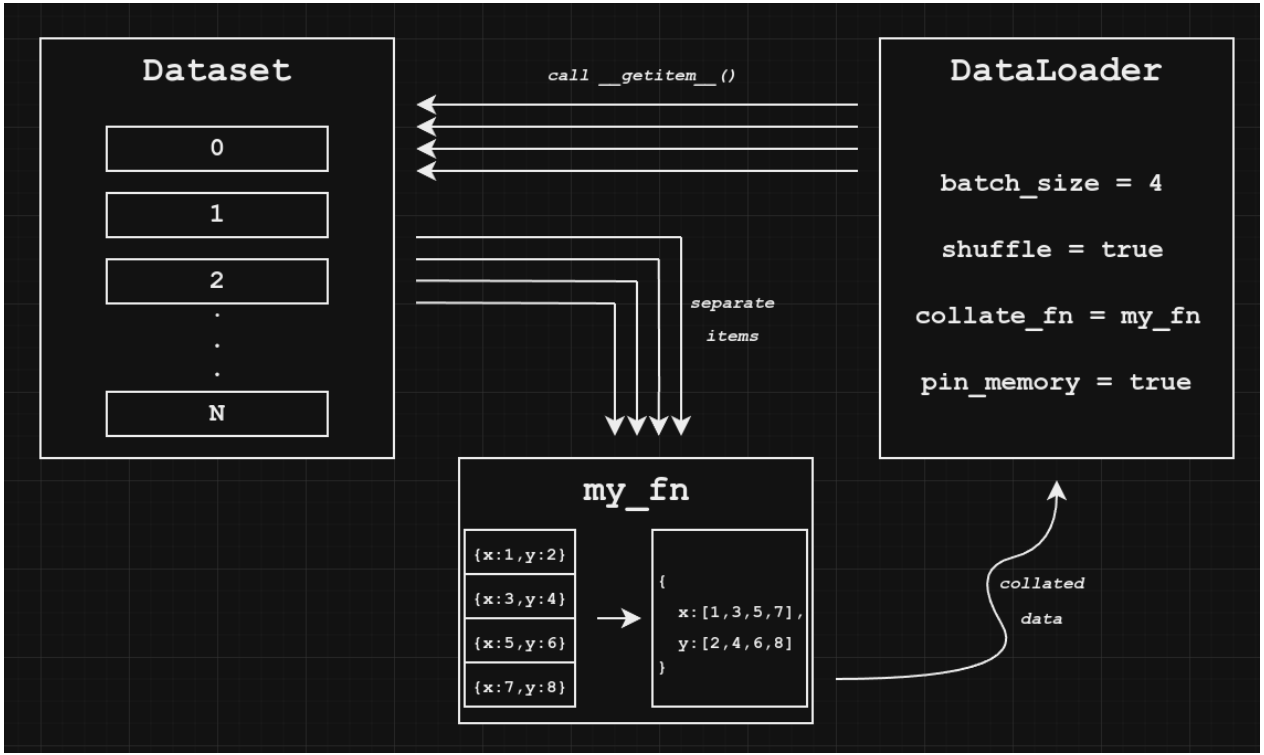
3. Dataloaders en Pytorch

Dataloaders

- Es una clase de diseñada para simplificar la carga y la iteración de conjuntos de datos durante el entrenamiento de modelos de aprendizaje profundo.
- Principales características:
  - *Carga las muestras de cada batch* para cada paso del entrenamiento, sin tener que cargar todas las muestas.
  - *Mezclar las muestras*: mezcla aleatoriamente los datos, evitando sesgos.
  - *Multiprocesamiento*: usa múltiples subprocesos para acelerar la carga.
  - *Customizable*: mediante la función collate_fn
  - *Iteración sencilla*: mediante loop for.

4. Funciones Collate

Funciones Collate



5. Técnicas avanzadas

Optimización de la Carga de Datos

- ***num_workers***: nos permite cargar los datos en diferentes subprocesos.
- ***prefetch_factor***: nos permite definir el número de batches que se cargan antes de ser necesarios.
- ***pin_memory***: fija la memoria de los tensores cargados en la RAM para transferencias más rápidas a la **GPU**.

6. EXTRA RESOURCES

Documentación

- https://pytorch.org/tutorials/beginner/basics/data_tutorial.html
- https://pytorch.org/vision/stable/datasets
- https://pytorch.org/tutorials/beginner/basics/transforms_tutorial.html

Deep Learning

Pytorch3 - Datasets, Dataloaders, Collate function

UNIVERSIDAD
POLITÉCNICA
DE MADRID

Máster
Deep Learning