Sử dụng với Spark

Tài liệu này giới thiệu nhanh về cách sử dụng Bộ dữ liệu với Spark, với trọng tâm cụ thể
về cách tải Spark DataFrame vào đối tượng Dataset.

Từ đó, bạn có thể truy cập nhanh vào bất kỳ phần tử nào và bạn có thể sử dụng nó làm trình tải dữ liệu để hu
các mô hình.

Tải từ Spark

Đối tượng Dataset là một trình bao bọc của bảng Mũi tên, cho phép đọc nhanh từ các mảng trong
tập dữ liệu cho các bộ căng PyTorch, TensorFlow và JAX.
Bảng Mũi tên là bộ nhớ được ánh xạ từ đĩa, có thể tải các tập dữ liệu lớn hơn bộ nhớ của bạn.
RAM có sẵn.

You can get a Dataset from a Spark DataFrame using Dataset.from_spark() :

```
>>> from datasets import Dataset
>>> df = spark.createDataFrame(
...data=[[1, "Elia"], [2, "Teo"], [3, "Fang"]],
...columns=["id", "name"],
... )
>>> ds = Dataset.from_spark(df)
```

Nhân viên Spark ghi tập dữ liệu vào đĩa trong thư mục bộ đệm dưới dạng tệp Mũi tên và
Tập dữ liệu được tải từ đó.

Alternatively, you can skip materialization by using IterableDataset.from_spark() , which
trả về một IterableDataset:

```
>>> from datasets import IterableDataset
>>> df = spark.createDataFrame(
...data=[[1, "Elia"], [2, "Teo"], [3, "Fang"]],
...columns=["id", "name"],
... )
>>> ds = IterableDataset.from_spark(df)
>>> print(next(iter(ds)))
{"id": 1, "name": "Elia"}
```

Bộ nhớ đệm

When using Dataset.from_spark() , the resulting Dataset is cached; if you call
Dataset.from_spark() multiple
nhiều lần trên cùng một DataFrame, nó sẽ không chạy lại công việc Spark ghi tập dữ liệu dưới dạng Mũi tên các tập tin trên đĩa.

You can set the cache location by passing cache_dir= to Dataset.from_spark() .
Make sure to use a disk that is available to both your workers and your current machine (the
driver).

> [!WARNING]
> Trong một phiên khác, Spark DataFrame không có cùng hàm băm ngữ nghĩa và nó sẽ chạy lại công việc Spark và lưu nó vào bộ đệm mới.

Các loại tính năng

Nếu tập dữ liệu của bạn được tạo từ hình ảnh, dữ liệu âm thanh hoặc mảng N chiều, bạn có thể chỉ định features= argument in
Dataset.from_spark() (or IterableDataset.from_spark()):

```
>>> from datasets import Dataset, Features, Image, Value
>>> data = [(0, open("image.png", "rb").read())]
>>> df = spark.createDataFrame(data, "idx: int, image: binary")
>>> # Also works if you have arrays
>>> # data = [(0, np.zeros(shape=(32, 32, 3), dtype=np.int32).tolist())]
>>> # df = spark.createDataFrame(data, "idx: int, image: array<array<array<int>>>")
>>> features = Features({"idx": Value("int64"), "image": Image()})
>>> dataset = Dataset.from_spark(df, features=features)
>>> dataset[0]
{'idx': 0, 'image': <PIL.PngImagePlugin.PngImageFile image mode=RGB size=32x32>}
```

Bạn có thể kiểm tra tài liệu Tính năng để biết về tất cả các loại tính năng có sẵn.