Tải dữ liệu dạng bảng

Tập dữ liệu dạng bảng là tập dữ liệu chung được sử dụng để mô tả bất kỳ dữ liệu nào được lưu trữ trong hàn
where the rows represent an example and the columns represent a feature (can be continuous
or categorical). These datasets are commonly stored in CSV files, Pandas DataFrames, and in
các bảng cơ sở dữ liệu. Hướng dẫn này sẽ chỉ cho bạn cách tải và tạo tập dữ liệu dạng bảng từ:

- Tệp CSV
- Khung dữ liệu Pandas
- Tập tin HDF5
- Cơ sở dữ liệu

Tệp CSV

Bộ dữ liệu có thể đọc tệp CSV bằng cách chỉ định tên trình tạo tập dữ liệu csv chung trong
load_dataset() method. To load more than one CSV file, pass them as a list to the data_files
tham số:

```
>>> from datasets import load_dataset
>>> dataset = load_dataset("csv", data_files="my_file.csv")

# tải nhiều tệp CSV
>>> dataset = load_dataset("csv", data_files=["my_file_1.csv", "my_file_2.csv", "my_file_3.csv"])
```

Bạn cũng có thể ánh xạ các tệp CSV cụ thể tới các phần tách thử nghiệm và đào tạo:

```
>>> dataset = load_dataset("csv", data_files={"train": ["my_train_file_1.csv", "my_train_file_2.cs
```

Để tải các tệp CSV từ xa, thay vào đó hãy chuyển các URL:

```
>>> base_url = "https://huggingface.co/datasets/lhoestq/demo1/resolve/main/data/"
>>> dataset = load_dataset('csv', data_files={"train": base_url + "train.csv", "test": base_url +
```

Để tải tệp CSV đã nén:

```
>>> url = "https://domain.org/train_data.zip"
>>> data_files = {"train": url}
>>> dataset = load_dataset("csv", data_files=data_files)
```

Khung dữ liệu gấu trúc

Datasets also supports loading datasets from Pandas DataFrames with the from_pandas()
phương pháp:

```
>>> from datasets import Dataset
>>> import pandas as pd

# tạo Khung dữ liệu Pandas
>>> df = pd.read_csv("https://huggingface.co/datasets/imodels/credit-card/raw/main/train.csv")
>>> df = pd.DataFrame(df)
# tải Bộ dữ liệu từ Pandas DataFrame
>>> dataset = Dataset.from_pandas(df)
```

Sử dụng tham số tách để chỉ định tên của phần tách tập dữ liệu:

```
>>> train_ds = Dataset.from_pandas(train_df, split="train")
>>> test_ds = Dataset.from_pandas(test_df, split="test")
```

Nếu tập dữ liệu trông không như mong đợi, bạn nên chỉ định rõ ràng các tính năng của tập dữ liệu. MỘT
pandas.Series có thể không phải lúc nào cũng mang đủ thông tin để Arrow tự động suy ra dữ liệu
kiểu. Ví dụ: nếu DataFrame có độ dài 0 hoặc nếu Sê-ri chỉ chứa Không/NaN
đối tượng, loại được đặt thành null .

tập tin HDF5

Các tệp HDF5 thường được sử dụng để lưu trữ lượng lớn dữ liệu số trong khoa học
tính toán và học máy. Tải tệp HDF5 bằng Bộ dữ liệu cũng tương tự như tải
Tệp CSV:

```
>>> from datasets import load_dataset
>>> dataset = load_dataset("hdf5", data_files="data.h5")
```

Lưu ý rằng trình tải HDF5 giả định rằng tệp có cấu trúc "dạng bảng", tức là tất cả các tập dữ liệu trong the file have (the same number of) rows on their first dimension.

Cơ sở dữ liệu

Các bộ dữ liệu được lưu trữ trong cơ sở dữ liệu thường được truy cập bằng các truy vấn SQL. Với Bộ dữ liệu, có thể kết nối với cơ sở dữ liệu, truy vấn dữ liệu bạn cần và tạo tập dữ liệu từ đó. Sau đó bạn có thể sử dụng tất cả các tính năng xử lý của Bộ dữ liệu để chuẩn bị tập dữ liệu cho việc đào tạo.

SQLite

SQLite là một cơ sở dữ liệu nhỏ, nhẹ, thiết lập nhanh chóng và dễ dàng. Bạn có thể sử dụng hiện có cơ sở dữ liệu nếu bạn muốn hoặc làm theo và bắt đầu lại từ đầu.

Bắt đầu bằng cách tạo cơ sở dữ liệu SQLite nhanh với dữ liệu Covid-19 này từ New York Times:

```
>>> import sqlite3
>>> import pandas as pd

>>> conn = sqlite3.connect("us_covid_data.db")
>>> df = pd.read_csv("https://raw.githubusercontent.com/nytimes/covid-19-data/master/us-states.csv
>>> df.to_sql("states", conn, if_exists="replace")
```

Thao tác này sẽ tạo một bảng trạng thái trong cơ sở dữ liệu us_covid_data.db mà giờ đây bạn có thể tải vào tập dữ liệu.

Để kết nối với cơ sở dữ liệu, bạn sẽ cần chuỗi URI xác định cơ sở dữ liệu của mình.
Việc kết nối với cơ sở dữ liệu bằng URI sẽ lưu trữ tập dữ liệu được trả về. Chuỗi URI khác nhau đối với từng phương ngữ cơ sở dữ liệu, vì vậy hãy nhớ kiểm tra URL cơ sở dữ liệu để biết bạn đang sử dụng cơ sở dữ sử dụng.

Đối với SQLite, đó là:

```
>>> uri = "sqlite:///us_covid_data.db"
```

Load the table by passing the table name and URI to from_sql():

```
>>> from datasets import Dataset

>>> ds = Dataset.from_sql("states", uri)
>>> ds
Dataset({
    features: ['index', 'date', 'state', 'fips', 'cases', 'deaths'],
    num_rows: 54382
})
```

Then you can use all of Datasets process features like filter() for example:

```
>>> ds.filter(lambda x: x["state"] == "California")
```

Bạn cũng có thể tải tập dữ liệu từ truy vấn SQL thay vì toàn bộ bảng, điều này rất hữu ích cho truy vấn và nối nhiều bảng.

Load the dataset by passing your query and URI to from_sql():

```
>>> from datasets import Dataset

>>> ds = Dataset.from_sql('SELECT * FROM states WHERE state="California";', uri)
>>> ds
Dataset({
    features: ['index', 'date', 'state', 'fips', 'cases', 'deaths'],
    số_hàng: 1019
})
```

Then you can use all of Datasets process features like filter() for example:

```
>>> ds.filter(lambda x: x["cases"] > 10000)
```

PostgreSQL

Bạn cũng có thể kết nối và tải tập dữ liệu từ cơ sở dữ liệu PostgreSQL, tuy nhiên chúng tôi sẽ không trực tiếp trình bày cách thực hiện trong tài liệu vì ví dụ này chỉ được chạy trong một cuốn sổ tay. Thay vào đó, hãy xem cách cài đặt và thiết lập máy chủ PostgreSQL trong phần này cuốn sổ!

After you've setup your PostgreSQL database, you can use the from_sql() method to load a tập dữ liệu từ một bảng hoặc truy vấn.