Biết tập dữ liệu của bạn

Có hai loại đối tượng tập dữ liệu, Tập dữ liệu thông thường và sau đó là IterableDataset . MỘT
Bộ dữ liệu cung cấp khả năng truy cập ngẫu nhiên nhanh chóng vào các hàng và ánh xạ bộ nhớ để tải đều bộ dữ liệu lớn chỉ sử dụng một lượng bộ nhớ thiết bị tương đối nhỏ. Nhưng đối với thực sự, thực sự lớn các bộ dữ liệu thậm chí không vừa trên đĩa hoặc trong bộ nhớ, IterableDataset cho phép bạn truy cập và sử dụng tập dữ liệu mà không cần đợi nó tải xuống hoàn toàn!

Hướng dẫn này sẽ chỉ cho bạn cách tải và truy cập Tập dữ liệu và IterableDataset.

Tập dữ liệu

Khi tải phần tách tập dữ liệu, bạn sẽ nhận được đối tượng Tập dữ liệu. Bạn có thể làm nhiều việc với một Đối tượng tập dữ liệu, đó là lý do tại sao việc học cách thao tác và tương tác với dữ liệu lại quan trọng được lưu trữ bên trong.

Hướng dẫn này sử dụng tập dữ liệu rotten_tomatoes, nhưng vui lòng tải bất kỳ tập dữ liệu nào bạn muốn và theo cùng!

```
>>> from datasets import load_dataset

>>> dataset = load_dataset("cornell-movie-review-data/rotten_tomatoes", split="train")
```

Lập chỉ mục

Tập dữ liệu chứa các cột dữ liệu và mỗi cột có thể là một loại dữ liệu khác nhau. các
chỉ mục hoặc nhãn trục được sử dụng để truy cập các ví dụ từ tập dữ liệu. Ví dụ: lập chỉ mục theo row trả về một từ điển của một ví dụ từ tập dữ liệu:

```
# Lấy hàng đầu tiên trong tập dữ liệu
>>> dataset[0]
{'label': 1,
 'text': 'tảng đá được mệnh danh là " conan " mới của thế kỷ 21 và anh ấy sẽ thực hiện
```

Sử dụng toán tử - để bắt đầu từ cuối tập dữ liệu:

```
# Lấy hàng cuối cùng trong tập dữ liệu
>>> dataset[-1]
{'label': 0,
 'text': 'mọi thứ thực sự trở nên kỳ lạ , mặc dù không đặc biệt đáng sợ : bộ phim đều có điềm báo và n
```

Lập chỉ mục theo tên cột trả về danh sách tất cả các giá trị trong cột:

```
>>> dataset["text"]
['the rock is destined to be the 21st century\'s new " conan " and that he\'s going to make a spla
 'Phần tiếp theo được xây dựng công phu và lộng lẫy của bộ ba phim "chúa tể của những chiếc nhẫn" lớn c
 'phim tiểu sử hiệu quả nhưng quá tẻ nhạt',
 None
 'mọi chuyện thực sự trở nên kỳ lạ, mặc dù không đặc biệt đáng sợ: bộ phim hoàn toàn mang tính điềm bá
```

Bạn có thể kết hợp lập chỉ mục tên hàng và cột để trả về một giá trị cụ thể tại một vị trí:

```
>>> dataset[0]["text"]
'tảng đá được mệnh danh là " conan " mới của thế kỷ 21 và anh ấy sẽ tạo ra một tấm khiên
```

Thứ tự lập chỉ mục không quan trọng. Lập chỉ mục theo tên cột trước tiên trả về một đối tượng Cột
bạn có thể lập chỉ mục như bình thường với các chỉ mục hàng:

```
>>> import time

>>> start_time = time.time()
>>> text = dataset[0]["text"]
>>> end_time = time.time()
>>> print(f"Elapsed time: {end_time - start_time:.4f} seconds")
Thời gian đã trôi qua: 0,0031 giây

>>> start_time = time.time()
>>> text = dataset["text"][0]
>>> end_time = time.time()
>>> print(f"Elapsed time: {end_time - start_time:.4f} seconds")
Thời gian đã trôi qua: 0,0042 giây
```

Cắt lát

Việc cắt lát trả về một lát - hoặc tập hợp con - của tập dữ liệu, rất hữu ích để xem một số hàng tại một lần. Để cắt một tập dữ liệu, hãy sử dụng toán tử : để chỉ định một phạm vi vị trí.

```
# Lấy ba hàng đầu tiên
>>> dataset[:3]
{'label': [1, 1, 1],
 'text': ['the rock is destined to be the 21st century\'s new " conan " and that he\'s going to ma
  'Phần tiếp theo được xây dựng công phu và lộng lẫy của bộ ba phim "chúa tể của những chiếc nhẫn" lớn
  'effective but too-tepid biopic']}

# Nhận các hàng từ ba đến sáu
>>> dataset[3:6]
{'label': [1, 1, 1],
 'text': ['if you sometimes like to go to the movies to have fun , wasabi is a good place to start
  "nổi lên như một điều gì đó hiếm có, một bộ phim vấn đề được quan sát một cách trung thực và sâu sắc
  'bộ phim cung cấp một số hiểu biết sâu sắc về tư duy loạn thần kinh của tất cả truyện tranh -- ngay cả nh
```

Bộ dữ liệu có thể lặp lại

Một IterableDataset được tải khi bạn đặt tham số phát trực tuyến thành True trong load_dataset():

```
>>> from datasets import load_dataset

>>> iterable_dataset = load_dataset("ethz/food101", split="train", streaming=True)
>>> for example in iterable_dataset:
...print(example)
...phá vỡ
{'image': <PIL.JpegImagePlugin.JpegImageFile image mode=RGB size=384x512 at 0x7F0681F5C520>, 'labe
```

Bạn cũng có thể tạo IterableDataset từ Tập dữ liệu hiện có, nhưng nó nhanh hơn phát trực tuyến mode vì tập dữ liệu được truyền trực tuyến từ các tệp cục bộ:

```
>>> from datasets import load_dataset

>>> dataset = load_dataset("cornell-movie-review-data/rotten_tomatoes", split="train")
>>> iterable_dataset = dataset.to_iterable_dataset()
```

IterableDataset lặp dần dần từng mẫu một tập dữ liệu tại một thời điểm, do đó bạn không phải đợi toàn bộ tập dữ liệu tải xuống trước khi bạn có thể sử dụng nó. Như bạn có thể tưởng tượng, điều nà khá hữu ích cho các tập dữ liệu lớn mà bạn muốn sử dụng ngay lập tức!

Lập chỉ mục

Hành vi của IterableDataset khác với Dataset thông thường. Bạn không có quyền truy cập ngẫu nhiên đến các ví dụ trong IterableDataset. Thay vào đó, bạn nên lặp lại các phần tử của nó, ví dụ: by calling next(iter()) or with a for loop to return the next item from the IterableDataset:

```
>>> next(iter(iterable_dataset))
{'image': <PIL.JpegImagePlugin.JpegImageFile image mode=RGB size=384x512 at 0x7F0681F59B50>,
  'label': 6}

>>> for example in iterable_dataset:
...print(example)
...phá vỡ
{'image': <PIL.JpegImagePlugin.JpegImageFile image mode=RGB size=384x512 at 0x7F7479DE82B0>, 'labe
```

Nhưng IterableDataset hỗ trợ lập chỉ mục cột trả về một giá trị có thể lặp lại cho các giá trị cột:

```
>>> next(iter(iterable_dataset["label"]))
6
```

Tạo một tập hợp con

Bạn có thể trả về một tập hợp con của tập dữ liệu với một số ví dụ cụ thể trong đó bằng IterableDataset.take():

```
# Lấy ba ví dụ đầu tiên
>>> list(iterable_dataset.take(3))
[{'image': <PIL.JpegImagePlugin.JpegImageFile image mode=RGB size=384x512 at 0x7F7479DEE9D0>,
  'label': 6},
 {'image': <PIL.JpegImagePlugin.JpegImageFile image mode=RGB size=512x512 at 0x7F7479DE8190>,
  'label': 6},
 {'image': <PIL.JpegImagePlugin.JpegImageFile image mode=RGB size=512x383 at 0x7F7479DE8310>,
  'label': 6}]
```

But unlike slicing, IterableDataset.take() creates a new IterableDataset.

Các bước tiếp theo

Bạn muốn tìm hiểu thêm về sự khác biệt giữa hai loại tập dữ liệu này? Học hỏi
tìm hiểu thêm về chúng trong hướng dẫn khái niệm Sự khác biệt giữa Tập dữ liệu và IterableDataset.

Để thực hành nhiều hơn với các loại tập dữ liệu này, hãy xem Hướng dẫn quy trình để tìm hiểu cách xử lý trước Tập dữ liệu hoặc Hướng dẫn luồng để tìm hiểu cách xử lý trước Tập dữ liệu lặp.