Ánh xạ hàng loạt

Combining the utility of Dataset.map() with batch mode is very powerful. It allows you to speed
xử lý và tự do kiểm soát kích thước của tập dữ liệu được tạo.

Cần tốc độ

Mục tiêu chính của ánh xạ hàng loạt là tăng tốc độ xử lý. Nhiều khi, sẽ nhanh hơn
làm việc với hàng loạt dữ liệu thay vì các ví dụ đơn lẻ. Đương nhiên, việc lập bản đồ hàng loạt có lợi cho
token hóa. Ví dụ: thư viện Tokenizers hoạt động nhanh hơn theo lô vì nó
song song việc mã hóa tất cả các ví dụ trong một đợt.

Input size != output size

Khả năng kiểm soát kích thước của tập dữ liệu được tạo có thể được tận dụng cho nhiều mục đích thú vị
trường hợp sử dụng. Trong phần Cách lập bản đồ, có các ví dụ về cách sử dụng bản đồ hàng loạt để:

- Chia những câu dài thành những đoạn ngắn hơn.
- Tăng cường tập dữ liệu bằng các mã thông báo bổ sung.

Sẽ rất hữu ích khi hiểu cách thức hoạt động của nó, để bạn có thể nghĩ ra cách riêng của mình để sử dụng
lập bản đồ hàng loạt. Tại thời điểm này, bạn có thể tự hỏi làm thế nào bạn có thể kiểm soát kích thước của
tập dữ liệu được tạo ra. Câu trả lời là: hàm được ánh xạ không phải trả về kết quả đầu ra
lô có cùng kích thước.

Nói cách khác, đầu vào hàm được ánh xạ của bạn có thể là một lô có kích thước N và trả về một lô có kích thu
M. Đầu ra M có thể lớn hơn hoặc nhỏ hơn N. Điều này có nghĩa là bạn có thể nối
ví dụ, chia nó ra và thậm chí thêm nhiều ví dụ khác!

Tuy nhiên, hãy nhớ rằng tất cả các giá trị trong từ điển đầu ra phải chứa cùng một số lượng
các phần tử như các trường khác trong từ điển đầu ra. Ngược lại, không thể xác định được
số lượng ví dụ trong đầu ra được hàm ánh xạ trả về. Số lượng có thể thay đổi
giữa các lô liên tiếp được xử lý bởi hàm ánh xạ. Tuy nhiên, đối với một lô duy nhất, tất cả
values of the output dictionary should have the same length (i.e., the number of elements).

Ví dụ: từ tập dữ liệu gồm 1 cột và 3 hàng, nếu bạn sử dụng bản đồ để trả về một cột mới

với số hàng gấp đôi thì bạn sẽ gặp lỗi.
Trong trường hợp này, bạn sẽ có một cột có 3 hàng và một cột có 6 hàng. Như bạn có thể xem, bảng sẽ không hợp lệ:

```
>>> from datasets import Dataset
>>> dataset = Dataset.from_dict({"a": [0, 1, 2]})
>>> dataset.map(lambda batch: {"b": batch["a"] * 2}, batched=True)# new column with 6 elements:
'ArrowInvalid: Cột 1 có tên b dự kiến có độ dài 3 nhưng lại có độ dài 6'
```

Để làm cho nó hợp lệ, bạn phải bỏ một trong các cột:

```
>>> from datasets import Dataset
>>> dataset = Dataset.from_dict({"a": [0, 1, 2]})
>>> dataset_with_duplicates = dataset.map(lambda batch: {"b": batch["a"] * 2}, remove_columns=["a"
>>> len(dataset_with_duplicates)
6
```

Ngoài ra, bạn có thể ghi đè lên cột hiện có để đạt được kết quả tương tự.
Ví dụ: đây là cách sao chép mọi hàng trong tập dữ liệu bằng cách ghi đè cột "a" :

```
>>> from datasets import Dataset
>>> dataset = Dataset.from_dict({"a": [0, 1, 2]})
# ghi đè cột "a" hiện có bằng các giá trị trùng lặp
>>> duplicated_dataset = dataset.map(
...lambda batch: {"a": [x for x in batch["a"] for _ in range(2)]},
...batched=True
... )
>>> duplicated_dataset
Dataset({
    features: ['a'],
    số_hàng: 6
})
>>> duplicated_dataset["a"]
[0, 0, 1, 1, 2, 2]
```