Sử dụng với NumPy

Tài liệu này là phần giới thiệu nhanh về cách sử dụng bộ dữ liệu với NumPy, đặc biệt tập trung vào
làm thế nào để có được
các đối tượng numpy.ndarray ra khỏi bộ dữ liệu của chúng tôi và cách sử dụng chúng để huấn luyện các mô
NumPy như mô hình scikit-learn.

Định dạng tập dữ liệu

Theo mặc định, bộ dữ liệu trả về các đối tượng Python thông thường: số nguyên, số float, chuỗi, danh sách, v

Thay vào đó, để nhận mảng NumPy, bạn có thể đặt định dạng của tập dữ liệu thành numpy :

```
>>> from datasets import Dataset
>>> data = [[1, 2], [3, 4]]
>>> ds = Dataset.from_dict({"data": data})
>>> ds = ds.with_format("numpy")
>>> ds[0]
{'data': array([1, 2])}
>>> ds[:2]
{'data': array([
       [1, 2],
       [3, 4]])}
```

[!TIP]
Đối tượng Dataset là một trình bao bọc của bảng Mũi tên, cho phép đọc nhanh từ các mảng trong
tập dữ liệu vào mảng NumPy.

Lưu ý rằng quy trình tương tự cũng áp dụng cho các đối tượng DatasetDict, do đó
khi đặt định dạng của DatasetDict thành numpy, tất cả các Dataset ở đó
sẽ được định dạng là numpy:

```
>>> from datasets import DatasetDict
>>> data = {"train": {"data": [[1, 2], [3, 4]]}, "test": {"data": [[5, 6], [7, 8]]}}
>>> dds = DatasetDict.from_dict(data)
>>> dds = dds.with_format("numpy")
>>> dds["train"][:2]
{'data': array([
        [1, 2],
        [3, 4]])}
```

Mảng N chiều

Nếu tập dữ liệu của bạn bao gồm các mảng N chiều, bạn sẽ thấy theo mặc định chúng là được coi là cùng một mảng nếu hình dạng được cố định:

```
>>> from datasets import Dataset
>>> data = [[[1, 2],[3, 4]], [[5, 6],[7, 8]]]# fixed shape
>>> ds = Dataset.from_dict({"data": data})
>>> ds = ds.with_format("numpy")
>>> ds[0]
{'data': array([[1, 2],
                [3, 4]])}
```

```
>>> from datasets import Dataset
>>> data = [[[1, 2],[3]], [[4, 5, 6],[7, 8]]]# varying shape
>>> ds = Dataset.from_dict({"data": data})
>>> ds = ds.with_format("numpy")
>>> ds[0]
{'data': array([array([1, 2]), array([3])], dtype=object)}
```

Tuy nhiên logic này thường yêu cầu so sánh hình dạng và sao chép dữ liệu chậm.
Để tránh điều này, bạn phải sử dụng rõ ràng loại tính năng Mảng và chỉ định hình dạng của tensor:

```
>>> from datasets import Dataset, Features, Array2D
>>> data = [[[1, 2],[3, 4]],[[5, 6],[7, 8]]]
>>> features = Features({"data": Array2D(shape=(2, 2), dtype='int32')})
>>> ds = Dataset.from_dict({"data": data}, features=features)
>>> ds = ds.with_format("numpy")
>>> ds[0]
{'data': array([[1, 2],
                [3, 4]])}
>>> ds[:2]
{'data': array([[[1, 2],
                 [3, 4]],

                [[5, 6],
                 [7, 8]]])}
```

Các loại tính năng khác

Dữ liệu ClassLabel được chuyển đổi chính xác thành mảng:

```
>>> from datasets import Dataset, Features, ClassLabel
>>> labels = [0, 0, 1]
>>> features = Features({"label": ClassLabel(names=["negative", "positive"])})
>>> ds = Dataset.from_dict({"label": labels}, features=features)
>>> ds = ds.with_format("numpy")
>>> ds[:3]
{'label': array([0, 0, 1])}
```

Các đối tượng chuỗi và nhị phân không thay đổi vì NumPy chỉ hỗ trợ số.

Các loại tính năng Hình ảnh và Âm thanh cũng được hỗ trợ.

> [!TIP]
> Để sử dụng loại tính năng Hình ảnh, bạn sẽ cần cài đặt thêm tầm nhìn như
> pip install datasets[vision] .

```
>>> from datasets import Dataset, Features, Image
>>> images = ["path/to/image.png"] * 10
>>> features = Features({"image": Image()})
>>> ds = Dataset.from_dict({"image": images}, features=features)
>>> ds = ds.with_format("numpy")
>>> ds[0]["image"].shape
(512, 512, 3)
>>> ds[0]
{'image': array([[[ 255, 255, 255],
                  [ 255, 255, 255],
                  None
                  [ 255, 255, 255],
                  [ 255, 255, 255]]], dtype=uint8)}
>>> ds[:2]["image"].shape
(2, 512, 512, 3)
>>> ds[:2]
{'image': array([[[[ 255, 255, 255],
                  [ 255, 255, 255],
                  None
                  [ 255, 255, 255],
                  [ 255, 255, 255]]]], dtype=uint8)}
```

[!TIP]
Để sử dụng loại tính năng Âm thanh, bạn sẽ cần cài đặt thêm âm thanh như
pip install datasets[audio] .

```
>>> from datasets import Dataset, Features, Audio
>>> audio = ["path/to/audio.wav"] * 10
>>> features = Features({"audio": Audio()})
>>> ds = Dataset.from_dict({"audio": audio}, features=features)
>>> ds = ds.with_format("numpy")
>>> ds[0]["audio"]["array"]
array([-0.059021, -0.03894043, -0.00735474, ...,0.0133667 ,
                  0.01809692,0.00268555], dtype=float32)
>>> ds[0]["audio"]["sampling_rate"]
array(44100, weak_type=True)
```

Đang tải dữ liệu

NumPy không có bất kỳ khả năng tải dữ liệu tích hợp nào, vì vậy bạn sẽ cần phải hiện thực hóa các mảng NumPy như X, y để sử dụng trong scikit-learn hoặc sử dụng thư viện như PyTorch để tải dữ liệu của bạn bằng cách sử dụng DataLoader .

Using with_format('numpy')

The easiest way to get NumPy arrays out of a dataset is to use the with_format('numpy') phương pháp. Hãy giả sử
rằng chúng tôi muốn đào tạo một mạng lưới thần kinh trên bộ dữ liệu MNIST có sẵn tại HuggingFace Hub tại https://huggingface.co/datasets/mnist.

```
>>> from datasets import load_dataset
>>> ds = load_dataset("mnist")
>>> ds = ds.with_format("numpy")
>>> ds["train"][0]
{'image': array([[0,0,0, ...],
                 [0,0,0, ...],
                 None
                 [0,0,0, ...],
                 [0,0,0, ...]], dtype=uint8),
  'label': array(5)}
```

Sau khi định dạng được đặt, chúng tôi có thể cung cấp tập dữ liệu cho mô hình dựa trên NumPy theo đợt bằ the Dataset.iter()
phương pháp:

```
>>> for epoch in range(epochs):
...for batch in ds["train"].iter(batch_size=32):
...x, y = batch["image"], batch["label"]
None
```