Bộ đệm

Bộ nhớ đệm là một trong những lý do khiến Bộ dữ liệu hoạt động hiệu quả. Nó lưu trữ trước đó các tập dữ liệu đã được tải xuống và xử lý để khi bạn cần sử dụng lại, chúng sẽ được tải lại trực tiếp từ bộ đệm. Điều này tránh phải tải xuống lại tập dữ liệu hoặc đăng ký lại các chức năng xử lý. Ngay cả sau khi bạn đóng và bắt đầu một phiên Python khác, Bộ dữ liệu sẽ tải lại tập dữ liệu của bạn trực tiếp từ bộ đệm!

Dấu vân tay

Bộ nhớ đệm theo dõi những biến đổi nào được áp dụng cho tập dữ liệu bằng cách nào? Vâng, Bộ dữ liệu gán dấu vân tay cho tệp bộ đệm. Dấu vân tay theo dõi trạng thái hiện tại của một tập dữ liệu. Dấu vân tay ban đầu được tính bằng cách sử dụng hàm băm từ bảng Mũi tên hoặc hàm băm các tệp Mũi tên nếu tập dữ liệu nằm trên đĩa. Dấu vân tay tiếp theo được tính toán bằng cách kết hợp dấu vân tay của trạng thái trước đó và hàm băm của biến đổi mới nhất được áp dụng.

> [!TIP]
> Biến đổi là bất kỳ phương pháp xử lý nào từ hướng dẫn Cách thực hiện, chẳng hạn như Dataset.map() or Dataset.shuffle().

Dấu vân tay thực tế trông như thế nào:

```
>>> from datasets import Dataset
>>> dataset1 = Dataset.from_dict({"a": [0, 1, 2]})
>>> dataset2 = dataset1.map(lambda x: {"a": x["a"] + 1})
>>> print(dataset1._fingerprint, dataset2._fingerprint)
d19493523d95e2dc 5b86abacd4b42434
```

Để một biến đổi có thể băm được, nó cần phải được chọn bằng thì là hoặc dưa chua.

Khi bạn sử dụng một phép biến đổi không thể băm, Bộ dữ liệu sẽ sử dụng dấu vân tay ngẫu nhiên và đưa ra cảnh báo. Biến đổi không thể băm được coi là khác với biến đổi trước đó biến đổi. Kết quả là Bộ dữ liệu sẽ tính toán lại tất cả các phép biến đổi. Hãy chắc chắn rằng các phép biến đổi có thể được tuần tự hóa bằng dưa chua hoặc thì là để tránh điều này!

Một ví dụ về thời điểm Bộ dữ liệu tính toán lại mọi thứ là khi bộ nhớ đệm bị tắt. Khi

điều này xảy ra, các tập tin bộ đệm được tạo mỗi lần và chúng được ghi vào một bộ đệm tạm thời thư mục. Khi phiên Python của bạn kết thúc, các tệp bộ đệm trong thư mục tạm thời sẽ được đã xóa. Một hàm băm ngẫu nhiên được gán cho các tệp bộ đệm này, thay vì dấu vân tay.

> [!TIP]
> When caching is disabled, use Dataset.save_to_disk() to save your transformed dataset hoặc nó sẽ bị xóa sau khi phiên kết thúc.

Băm

Dấu vân tay của tập dữ liệu được cập nhật bằng cách băm hàm được truyền vào bản đồ cũng như map parameters (batch_size , remove_columns , etc.).

Bạn có thể kiểm tra hàm băm của bất kỳ đối tượng Python nào bằng dấu vân tay.Hasher:

```
>>> from datasets.fingerprint import Hasher
>>> my_func = lambda example: {"length": len(example["text"])}
>>> print(Hasher.hash(my_func))
'3d35e2b3e94c81d6'
```

Giá trị băm được tính bằng cách loại bỏ đối tượng bằng cách sử dụng dụng cụ nhặt thì là và băm đối tượng b byte.
Bộ chọn sẽ loại bỏ đệ quy tất cả các biến được sử dụng trong hàm của bạn, do đó, bất kỳ thay đổi nào bạn th một đối tượng được sử dụng trong hàm của bạn sẽ khiến hàm băm thay đổi.

Nếu một trong các hàm của bạn dường như không có cùng hàm băm trong các phiên, điều đó có nghĩa là ít một trong các biến của nó chứa một đối tượng Python không mang tính quyết định.
Khi điều này xảy ra, hãy thoải mái băm bất kỳ đối tượng nào bạn thấy nghi ngờ để cố gắng tìm ra đối tượng khiến hàm băm thay đổi.
Ví dụ: nếu bạn sử dụng một danh sách mà thứ tự các phần tử của nó không được xác định trên toàn bộ phiên thì hàm băm cũng sẽ không giống nhau giữa các phiên.