

직렬화

2015년 10월 26일에 한국어로 옮겨짐

원문: <https://github.com/torch/torch7/blob/master/doc/serialization.md>

목차

토치(Torch)는 임의의 루아/토치 객체들을 직렬화 및 직렬화 해제하기 위한 네 개의 고수준 메소드들을 제공합니다. 이 함수들은 단지 파일 객체의 추상들이고 편의를 위해 생성됩니다 (이것들은 매우 흔한 루틴들입니다).

첫 두 함수는 파일들에서 또는 파일로 데이터를 직렬화 및 직렬화 해제 하는 데 유용합니다.

- `torch.save(filename, object [, format, referenced])`
- `[object] torch.load(filename [, format, referenced])`

다음 두 함수는 문자열에서 또는 문자열로 데이터를 직렬화 및 직렬화 해제 하는 데 유용합니다.

- `[str] torch.serialize(object)`
- `[object] torch.deserialize(str)`

파일들로의 직렬화는 임의의 자료 구조를 저장하는 데 또는 그것들을 다른 사람들과 공유하는 데 유용합니다. 문자열로의 직렬화는 임의의 자료 구조 또는 제 3의 소프트웨어를 저장하는 데 유용합니다.

`torch.save(filename, object [, format, referenced])`

`object`를 `filename`이라는 이름의 한 파일로 씁니다. 그 `format`은 `ascii` 또는 `binary`로 설정될 수 있습니다 (기본값은 바이너리). 바이너리 포맷은 플랫폼에 의존적입니다. 그러나 보통 읽기/쓰기 시 더 간결하고 빠릅니다. 아스키 포맷은 플랫폼 독립적입니다. 그리고 여러 플랫폼 사이에 자료 구조를 공유할 때 꼭 사용되어야 합니다. 옵션 `referenced`는 객체 참조들이 추적되어야 하는지 아닌지를 특정합니다 (기본값은 `true`).

```
-- 임의의 객체:
obj = {
  mat = torch.randn(10,10),
  name = '10',
  test = {
```

```

    entry = 1
  }
}

-- 디스크로 저장:
torch.save('test.dat', obj)

```

[object] torch.load(filename [, format, referenced])

object를 filename이라는 이름의 한 파일에서 읽습니다. 그 format은 ascii 또는 binary로 설정될 수 있습니다 (기본값은 바이너리). 바이너리 포맷은 플랫폼에 의존적입니다. 그러나 보통 읽기/쓰기 시 더 간결하고 빠릅니다. 아스키 포맷은 플랫폼 독립적입니다. 그리고 여러 플랫폼 사이에 자료 구조를 공유할 때 꼭 사용되어야 합니다. 옵션 referenced는 [객체 참조](#)들이 추적되어야 하는지 아닌지를 특정합니다 (기본값은 true). 유념하십시오. referenced가 true인 상태에서 쓰여진 파일들은 referenced가 false인 상태에서 로드될 수 없습니다.

```

-- 위 절에서 직렬화된 객체를 다시 로드:
obj = torch.load('test.dat')

print(obj)
-- 다음을 출력할 것입니다:
-- {[mat] = DoubleTensor - size: 10x10
-- [name] = string : "10"
-- [test] = table - size: 0}

```

[str] torch.serialize(object [, format])

object를 한 문자열로 직렬화합니다. 그 format은 ascii 또는 binary로 설정될 수 있습니다 (기본값은 바이너리). 바이너리 포맷은 플랫폼에 의존적입니다. 그러나 보통 읽기/쓰기 시 더 간결하고 빠릅니다. 아스키 포맷은 플랫폼 독립적입니다. 그리고 여러 플랫폼 사이에 자료 구조를 공유할 때 꼭 사용되어야 합니다.

```

-- 임의의 객체:
obj = {
  mat = torch.randn(10,10),
  name = '10',
  test = {

```

```
    entry = 1
  }
}
-- 직렬화:
str = torch.serialize(obj)
```

[object] torch.deserialize(str [, format])

한 문자열에서 object를 직렬화-해제합니다. 그 format은 ascii 또는 binary로 설정될 수 있습니다 (기본값은 바이너리). 바이너리 포맷은 플랫폼에 의존적입니다. 그러나 보통 읽기/쓰기 시 더 간결하고 빠릅니다. 아스키 포맷은 플랫폼 독립적입니다. 그리고 여러 플랫폼 사이에 자료 구조를 공유할 때 꼭 사용되어야 합니다.

```
-- 위 절에서 직렬화한 객체를 직렬화-해제:
obj = torch.deserialize(str)

print(obj)
-- 다음을 출력할 것입니다:
-- {[mat] = DoubleTensor - size: 10x10
-- [name] = string : "10"
-- [test] = table - size: 0}
```

목차

[torch.save\(filename, object \[, format, referenced\]\)](#)
[\[object\] torch.load\(filename \[, format, referenced\]\)](#)
[\[str\] torch.serialize\(object \[, format\]\)](#)
[\[object\] torch.deserialize\(str \[, format\]\)](#)

[목차](#)

❖ 틀렸거나 보완할 점을 본문에 댓글로 또는 저에게 [이메일](#)로 알려 주시면 감사하겠습니다.