

STEP31. 고차미분 (이론 편)

문제의 핵심

- 계산 그래프의 연결이 만들어지는 시점 == 순전파를 계산할 때 만들어짐
- 역전파를 계산할 때는 연결이 만들어지지 않음

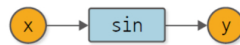
역전파 계산

역전파시의 계산 그래프

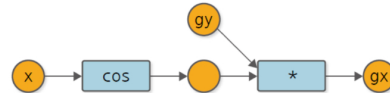
- 현재 구현된 backward 메서드는 계산만 진행되고 계산과 관련된 아무런 계산 그래프도 만들지 않음 (계산에서는 ndarray 인스턴스가 사용되기 때문)

고차 미분을 자동으로 계산할 수 있는 아이디어

- 역전파를 계산할 때도 연결이 만들어 지도록 하면 됨



- sin 함수의 미분을 구하기 위한 계산 그래프

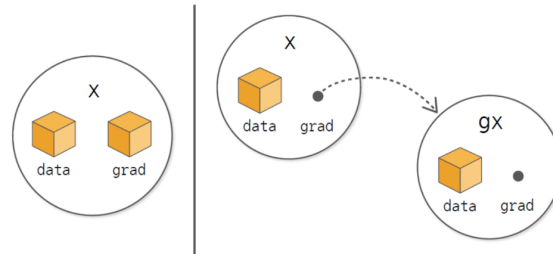


- `gx.backward()` 를 호출하여 `gx`의 `x`에 대한 미분을 계산할 수 있음
- `gx`는 `y = sin(x)`의 미분이기 때문에 `gx.backward()`를 호출함으로써 `x`의 2차 미분에 해당함

역전파로 계산 그래프 만들기

순전파 계산의 연결

- Variable 인스턴스를 사용하여 순전파를 하는 시점에서 연결이 만들어짐
- `backward()` 메서드에서 ndarray 인스턴스가 아닌 Variable 인스턴스를 사용하면 계산의 연결이 만들어진다는 뜻임
- 미분값(기울기)를 Variable 인스턴스 형태로 유지해야 함
- Variable 클래스의 grad는 ndarray 인스턴스를 참조하는 대신 Variable 인스턴스를 참조하도록 변경



Sin 클래스의 순전파와 역전파의 계산 그래프

- Variable 클래스의 grad가 Variable 인스턴스를 참조
- 미분값을 나타내는 `gy`가 Variable 인스턴스가 된 덕분에 `gy`를 사용한 계산에도 연결이 만들어짐
- Sin 클래스에서 `backward()` 메서드 구현시 미분을 계산하는 코드 == `gx = gy * cos(x)`

