

STEP30. 고차미분 (준비 편)

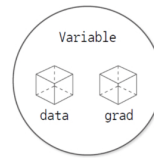
모든 고차미분 자동으로 계산

- N차 미분까지 자동으로 계산시 고려 사항
 - 현재의 역전파 구현 근본적으로 재검토 필요
 - 현재의 Variable과 Function 클래스 구현에 대한 고찰 필요

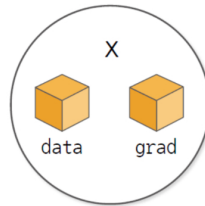
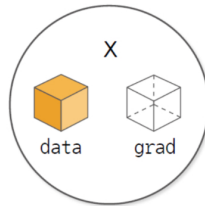
확인 i : Variable 인스턴스 변수

Variable 클래스의 init 메서드

- data와 grad 인스턴스 변수는 각각 순전파 계산과 역전파 계산 시 사용
- data와 grad 모두 ndarray 인스턴스를 저장함
(data와 grad는 입방체 상자로 그림)



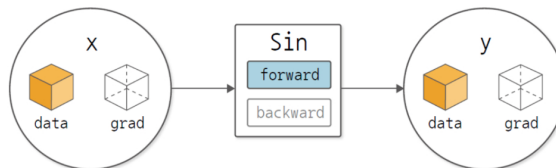
- data와 grad가 ndarray 인스턴스를 참조하는 경우



확인 ii : Function 클래스

Function 클래스의 call 메서드

- 순전파 계산(메인 처리)
 - `xs = [x.data for x in inputs]`
 - `forward(*xs)`를 호출하여 구체적인 계산
- Variable과 Function의 '관계'가 만들어짐
 - 변수에서 함수로의 연결은 `set_creator` 메서드가 만들어줌
 - 변수와 함수의 연결을 만드는 이유는 미분값을 역방향으로 진행하기 때문임
- Sin 클래스 코드
 - 순전파의 구체적인 계산은 sin 클래스의 forward 메서드에서 진행
 - 변수와 함수의 연결이 만들어짐, 연결은 `__call__` 메서드에서 만들어짐



확인 iii : Variable 클래스의 역전파

역전파 로직

- 역전파는 Variable 클래스의 backward 메서드에서 구현함
 - Variable의 인스턴스 변수인 grad를 리스트로 모음
 - backward 메서드에는 ndarray 인스턴스가 담긴 리스트가 전달됨
 - 출력 쪽에서 전파하는 미분값(gxs)을 함수의 입력 변수 (`f.inputs`)의 grad로 설정함
- 변수와 함수의 동작을 시각화
 - 순전파 $y = \sin(x)$ 가 실행될 때 계산 그래프가 만들어지고, Variable 인스턴스 변수인 data가 채워짐
 - 역전파 시 Sin 클래스의 backward 메서드가 불리고 Variable의 인스턴스 변수인 grad가 채워짐

