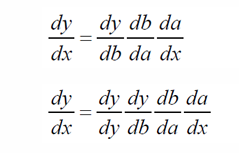
3주차 정리

1. **오차역전파법(back propagation)**

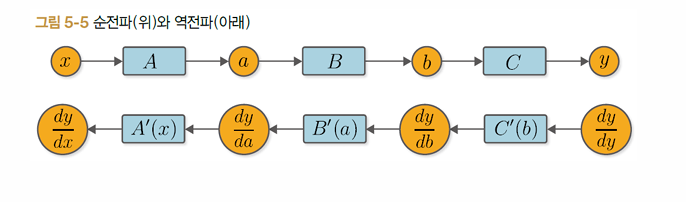
: 수치 미분보다 계산량과 오차 면에서 우수함

1. **역전파를 이해하려면 연쇄법칙부터 알아보자**

합성함수의 계산으로 설명하자면



위와 같은 합성함수의 미분은 구성 함수들의 미분의 곱으로 분해할 수 있음을 말한다.

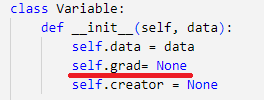


* 역전파를 구하려면 먼저 순전파를 하고, 이때 각 함수가 입력 변수의 값을 기억해야 함

1. **수동 역전파**

* 역전파를 이해하기 위해 역전파를 이용한 미분 구현을 해보자

우선 역전파에 대응하는 Variable 클래스 구현

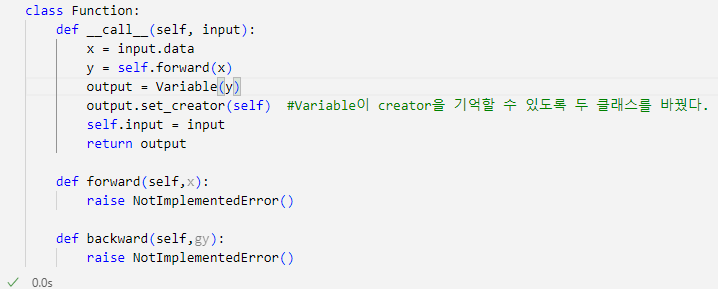
:

* grad 라는 인스턴스 변수 추가

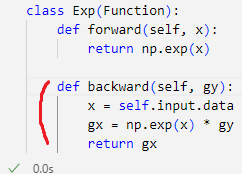
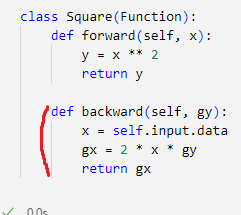
* grad는 역전파 수행시

미분값을 계산하여 대입

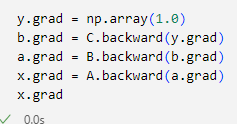
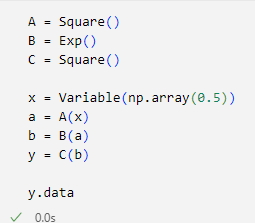
* Function 클래스에 추가 기능



* Square와 Exp 클래스 추가 구현



* 역전파를 담당하는 backward 메서드 추가
* **순전파 계산과 역전파 계산 비교**



* C -> B -> A 순으로 backward 호출

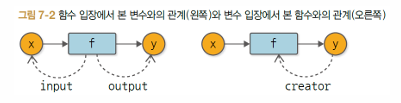
1. **역전파 자동화**

: 하나하나 계산하는게 힘드니 역전파를 자동화시킨다.

* define-by-Run

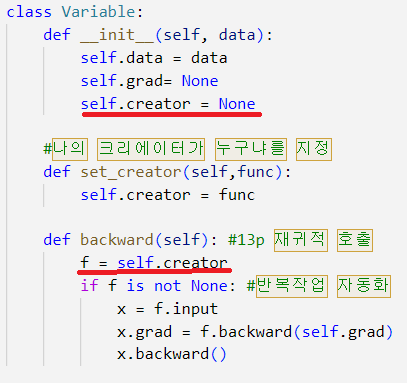
: 텐서플로우2에서는 순전파 계산이 동적인데 얘는 자동으로 그려지는 구조

* 방법 1: 변수와 함수의 ‘관계’

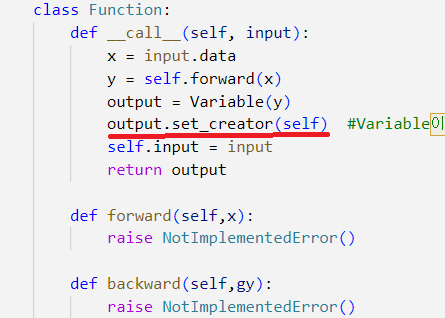


코드

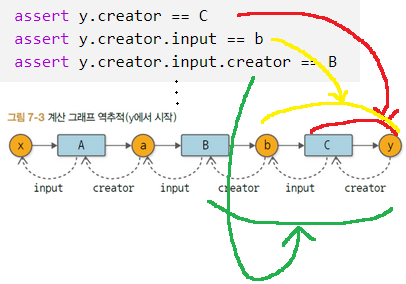
:



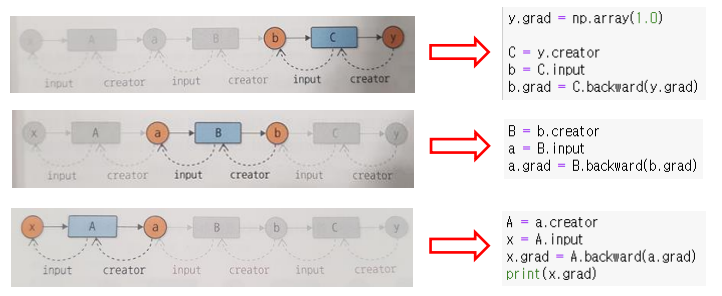
* creator 추가



* ‘연결’된 Variable과 Function 으로 계산 그래프를 거꾸로 거슬러 올라갈 수 있음

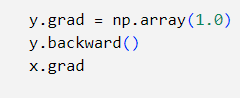


* assert 문으로 조건 충족 여부 확인
* 변수와 함수의 관계를 이용한 역전파 코드



1. **반복작업 자동화**

* 재귀적 호출



* 반복적 호출 구현

