

STEP34. sin 함수 고차 미분

새로운 DeZero 함수 추가

- 고차 미분에 대응하는 함수들을 core.py에 구현함
 - Add, Mul, Neg, Sub, Div, Pow 클래스
- 새로운 함수는 dezero/functions.py에 추가함
- 새로운 함수를 다른 파일에서 사용시
 - from dezero.functions import sin 형태로 임포트 해야함

sin 함수 구현

```
import numpy as np
from dezero.core import Function

class Sin(Function):
    def forward(self, x):
        y = np.sin(x)
        return y

    def backward(self, gy):
        x, = self.inputs
        gx = gy * cos(x)
        return gx

def sin(x):
    return Sin()(x)
```

고차 미분에 대응하는 새로운 Sin 클래스 구현

- 수식 $y = \sin(x)$ 일때 $y' = \cos(x)$
- backward 메서드 안의 모든 변수가 Variable 인스턴스임
- $gx = gy * \cos(x)$ 에서 $\cos(x)$ 는 DeZero의 cos 함수임
 - Sin 클래스를 구현하려면 Cos 클래스와 cos 함수가 필요함
- $gy * \cos(x)$ 에는 곱셈 연산자를 오버로드해 놓았기 때문에 mul 함수가 호출됨

cos 함수 구현

고차 미분에 대응하는 새로운 Cos 클래스와 cos 함수 구현

```
class Cos(Function):
    def forward(self, x):
        y = np.cos(x)
        return y

    def backward(self, gy):
        x, = self.inputs
        gx = gy * -sin(x)
        return gx

def cos(x):
    return Cos()(x)
```

- 수식 $y = \cos(x)$ 일때 $y' = -\sin(x)$
- backward 메서드에서 구체적인 계산에서 sin 함수를 사용

sin 함수 고차 미분

sin 함수의 고차 미분

```
import dezero.functions as F

x = Variable(np.array(1.0))
y = F.sin(x)
y.backward(create_graph=True)

for i in range(3):
    gx = x.grad()
    x.cleargrad()
    gx.backward(create_graph=True)
    print(x.grad)
```

- 2차 미분뿐 아니라 3차 미분, 4차 미분도 계산
- for 문을 사용하여 역전파를 반복하여, n차 미분을 구함

결과: 1차 미분값은 0.7071067811865475, 2차 미분값은 -0.7071067811865475, 3차 미분값은 0.7071067811865475

- 먼저 $gx = x.grad$ 에서 미분값을 꺼내 gx 에서 역전파하는 것임
- 역전파를 하기 전에 $x.cleargrad()$ 를 호출하여 미분값을 재설정함
- 이 작업을 반복하여 n 차 미분을 계산

sin 함수의 고차 미분 그래프 그리기

