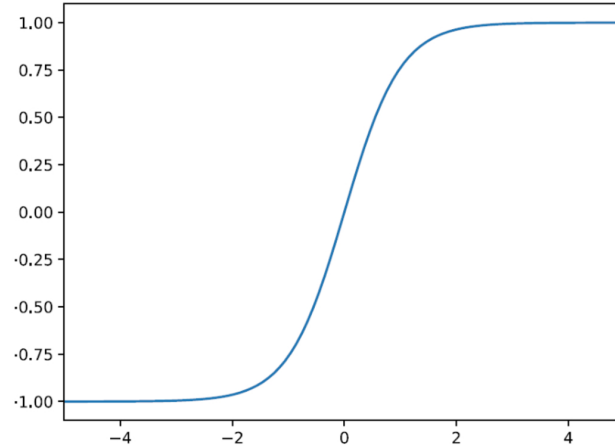


STEP35. 고차 미분 계산 그래프

tanh 함수 추가

- tanh 는 쌍곡탄젠트 혹은 하이퍼볼릭 탄젠트로 읽음
 - 수식은
$$y = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$
- tanh 함수는 입력을 -1 ~ 1 사이의 값으로 변환



tanh 함수 미분

tanh 함수 미분

- tanh 함수의 미분 공식을 이용해 계산

$$\left\{ \frac{f(x)}{g(x)} \right\}' = \frac{f'(x)g(x) - f(x)g'(x)}{g(x)^2}$$

- 분수 함수의 미분 공식을 이용하여 tanh 함수는 다음과 같이 미분할 수 있음
 - 최종적으로 $1 - y^2$ 이라는 결과를 얻음

$$\begin{aligned} \frac{\partial \tanh(x)}{\partial x} &= \frac{(e^x + e^{-x})(e^x + e^{-x}) - (e^x - e^{-x})(e^x - e^{-x})}{(e^x + e^{-x})^2} \\ &= 1 - \frac{(e^x - e^{-x})(e^x - e^{-x})}{(e^x + e^{-x})^2} \\ &= 1 - \left\{ \frac{(e^x - e^{-x})}{(e^x + e^{-x})} \right\}^2 \\ &= 1 - \tanh(x)^2 \\ &= 1 - y^2 \end{aligned}$$

tanh 함수 구현

tanh 함수의 미분

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

    def backward(self, gy):
        x_ = self.outputs[0]()
        gx = gy * (1 - y * y)
        return gx

def tanh(x):
    return Tanh()(x)
```

- $y = \tanh(x)$ 일때, $y' = 1 - y^2$ 임.
- 순전파에서 np.tanh 메서드를 이용함
- 역전파에서는 $gy * (1 - y * y)$ 형태로 구현
- 재사용할 수 있도록 dezero/functions.py에 추가

고차 미분 계산 그래프 시각화

고차 미분 계산 그래프 시각화

```
from dezero.utils import plot_dot_graph

x = Variable(np.array(1.0))
y = F.tanh(x)
x.name = 'x'
y.name = 'y'
y.backward(create_graph=True)

iters = 0

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

gx = x.grad
gx.name = 'gx' + str(iters + 1)
plot_dot_graph(gx, verbose=False, to_file='tanh.png')
```

- for 문에서 반복해서 역전파함으로 고차 미분을 계산
- iters = 0 이면 1차 미분, 1이면 2차 미분이 계산되는 방식임

