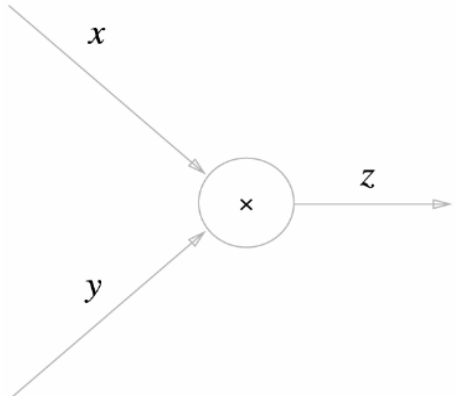
5.4 계층 구현하기

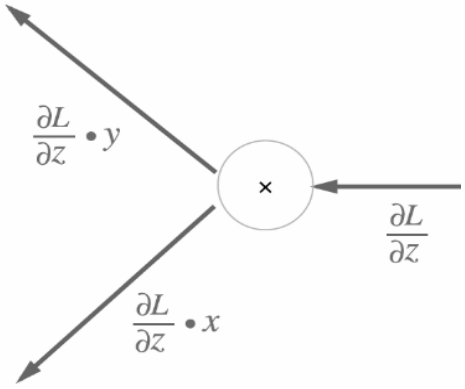
1. 곱셈 계층 (MulLayer)

```
class MulLayer:
    def __init__(self):
        self.x = None
        self.y = None

    def forward(self,x,y):
        self.x = x
        self.y = y
        out = x*y
        return x*y

    def backward(self, dout):
        dx = doub*self.y # x와 y 바꿈
        dy = doub*self.x
        return dx, dy
```



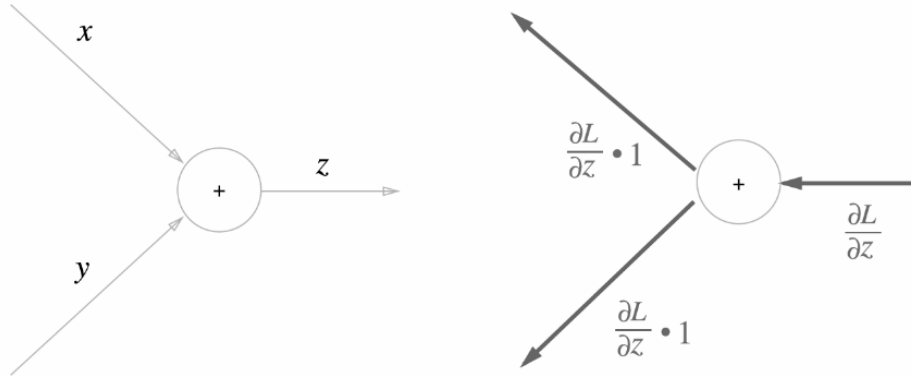


2. 덧셈 계층 (AddLayer)

```
class AddLayer:
    def __init__(self):
        pass

    def forward(self,x,y):
        out = x+y
        return out

    def backward(self,dout):
        dx = dout*1
        dy = dout*1
        return dx, dy
```



5.5 활성화 함수 계층 구현하기

1. ReLU 계층

$$y = \begin{cases} x & (x > 0) \\ 0 & (x \le 0) \end{cases} \quad \rightarrow \quad \frac{\partial y}{\partial x} = \begin{cases} 1 & (x > 0) \\ 0 & (x \le 0) \end{cases}$$


```
class Relu:
    def __init__(self):
        self.mask = None # mask는 True/False로 구성된 넘파이 배열
                         # 순전파 입력 x <= 0이면 True, x > 0이면 False

    def forward(self,x):
        self.mask = (x<=0)
        out = x.copy()
        out[self.mask]=0
        return out

    def backward(self, dout):
        dout[self.mask]=0
        dx = dout
        return dx
```