5강: 역전파

개요

- 함수 f(x)에 대해 x에서의 f의 그래디언트($\nabla f(x)$)를 계산하는 것이 목표
- 함수 f는 손실 함수이며, 훈련 데이터 (x_i, y_i) $(i = 1, \dots, N)$ 및 가중치 W, 편향벡터 b의 함수일 수 있음
- 파라미터 W, b에 대한 그래디언트를 계산한 후, 파라미터 업데이트 시 사용

그래디언트 예시

- 곱셈 함수

$$f(x,y) = xy \qquad \rightarrow \qquad \frac{\partial f}{\partial x} = y \qquad \frac{\partial f}{\partial y} = x$$

- 덧셈 함수

$$f(x,y) = x + y \qquad \rightarrow \qquad \frac{\partial f}{\partial x} = 1 \qquad \frac{\partial f}{\partial y} = 1$$

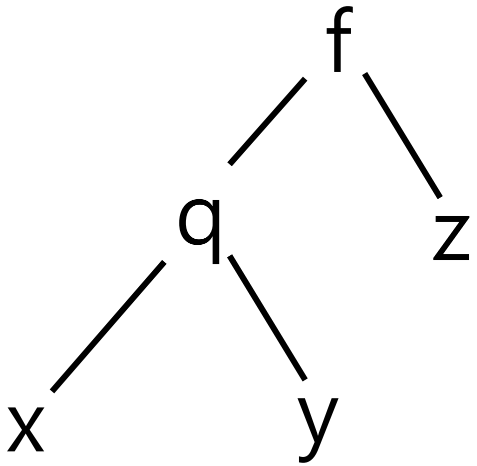
- 최대값 함수

$$f(x,y) = \max(x,y) \qquad \rightarrow \qquad \frac{\partial f}{\partial x} = 1(x >= y) \qquad \frac{\partial f}{\partial y} = 1(y >= x)$$

  – 그래디언트가 더 큰 입력에서는 1이고, 다른 입력에서는 0

연쇄 법칙 사용

- f(x,y,z) = (x+y)z
  - q=x+y, f=qz 로 표현

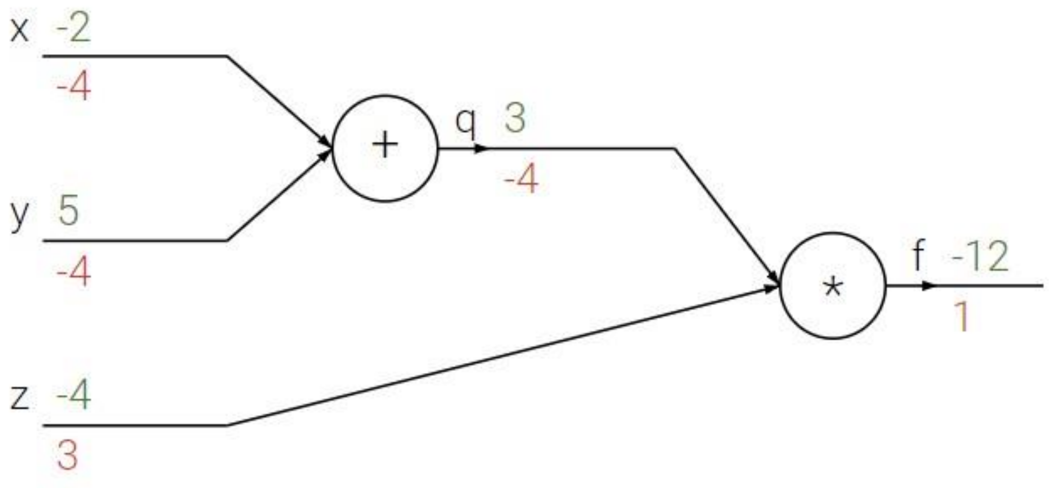


$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q}\frac{\partial q}{\partial x}$$

- df/dq = z, dq/dx = 1이므로 df/dx = z
- x=-2, y=5, z=-4에서 df/dx = -4
- df/dy = df/dq dq/dy = z = -4
- df/dz = q = x+y = 3

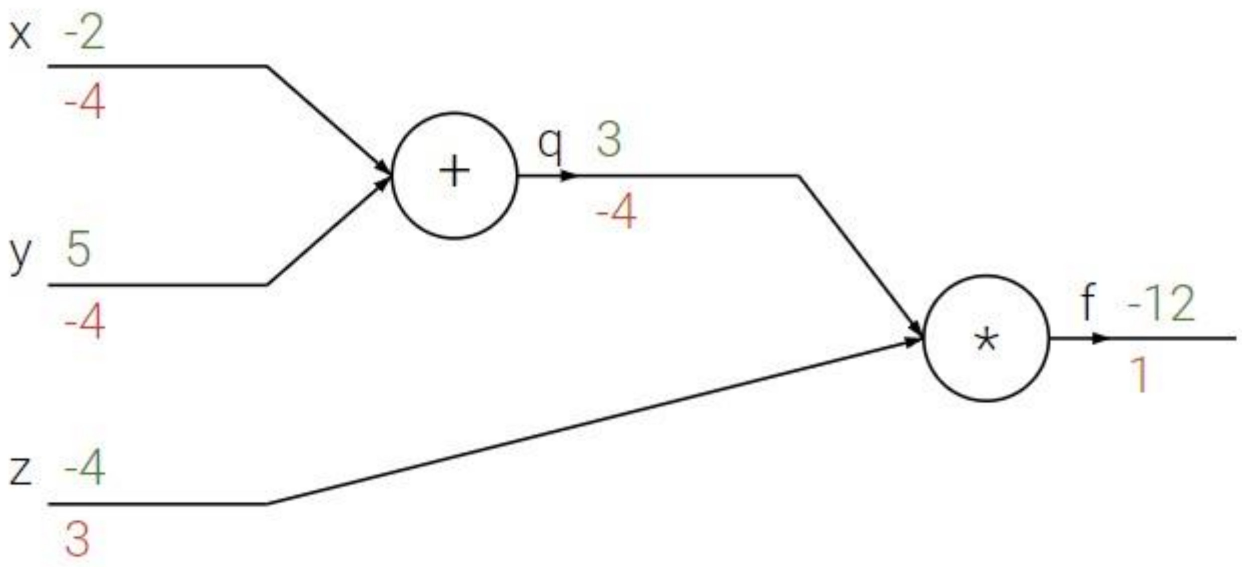
연쇄 법칙 사용

- df/dx, df/dy, df/dz는 x,y,z가 f에 미치는 민감도를 알려줌
- df/dx 대신 dx로 간단하게 표기



- 역전파는 지역적으로 수행됨
- 각 게이트는 입력을 받고, (1) 출력 값과 (2) 그 출력에 대한 입력에 대한 그래디언트를 바로 계산

역전파 직관



Sigmoid 함수 역전파 예시

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}} \qquad \sigma(x) = \frac{1}{1+e^{-x}}$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

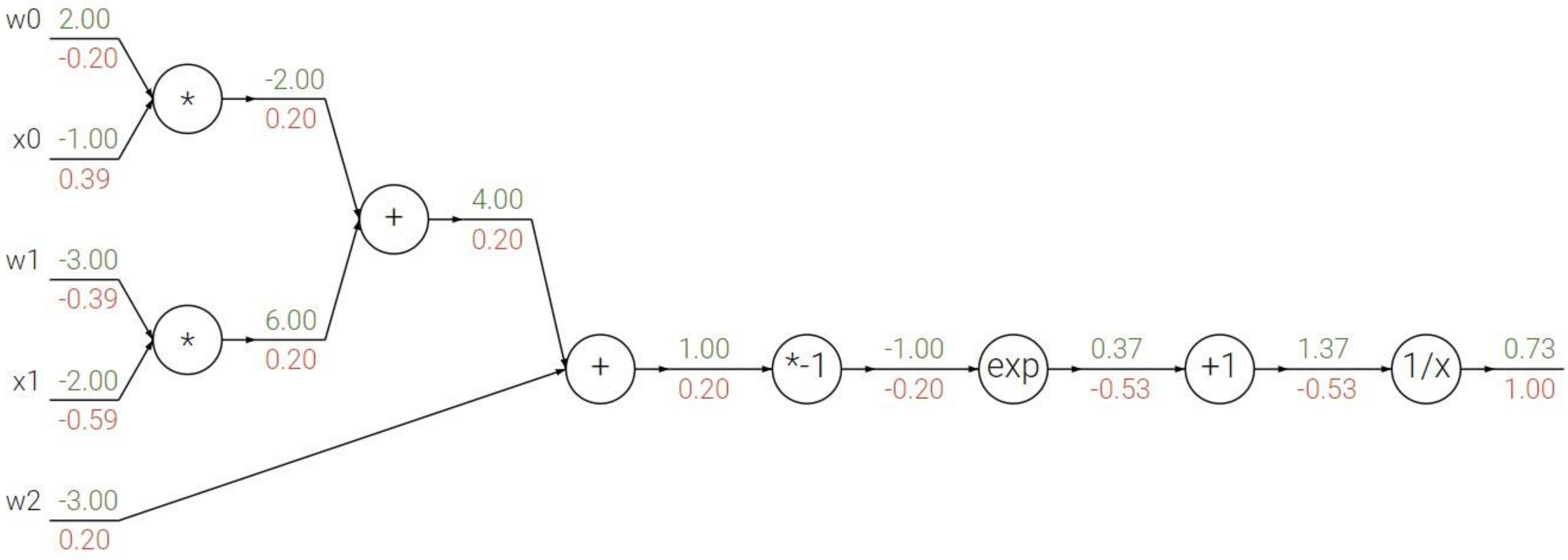
$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

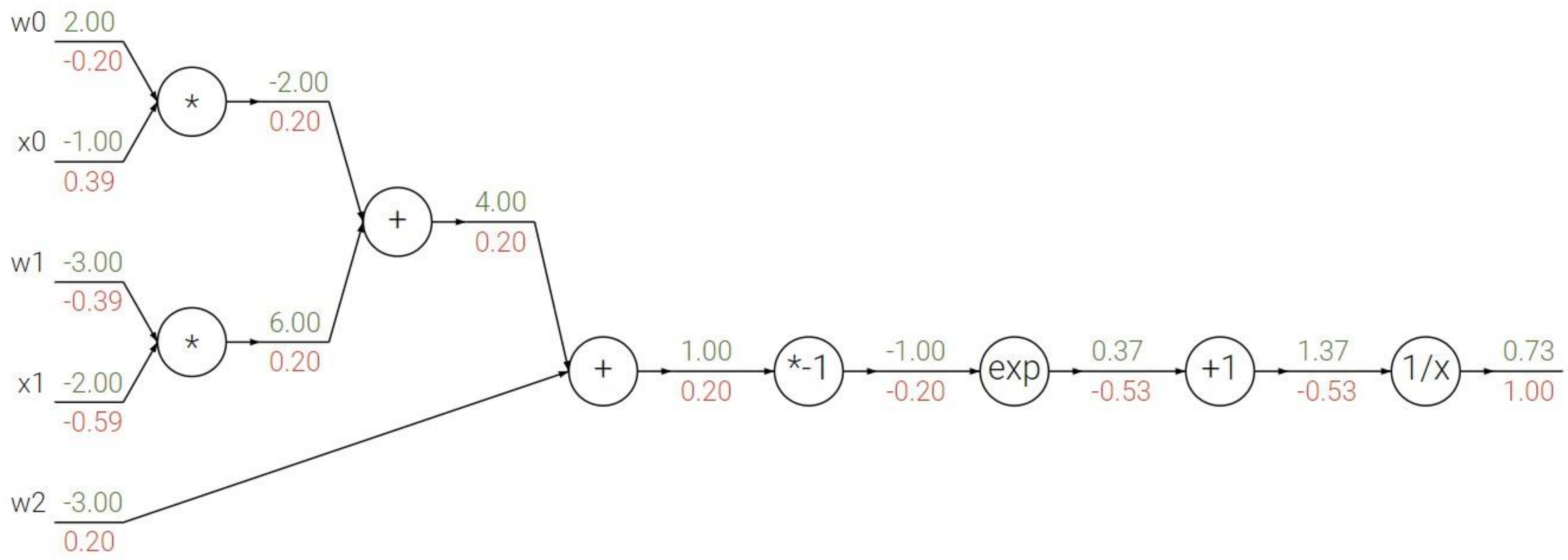
$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

Sigmoid 함수 역전파 예시

$$\frac{d\sigma(x)}{dx} = \frac{e^{-x}}{(1+e^{-x})^2} = \left(\frac{1+e^{-x}-1}{1+e^{-x}}\right)\left(\frac{1}{1+e^{-x}}\right) = (1-\sigma(x))\,\sigma(x)$$



Sigmoid 함수 역전파 예시



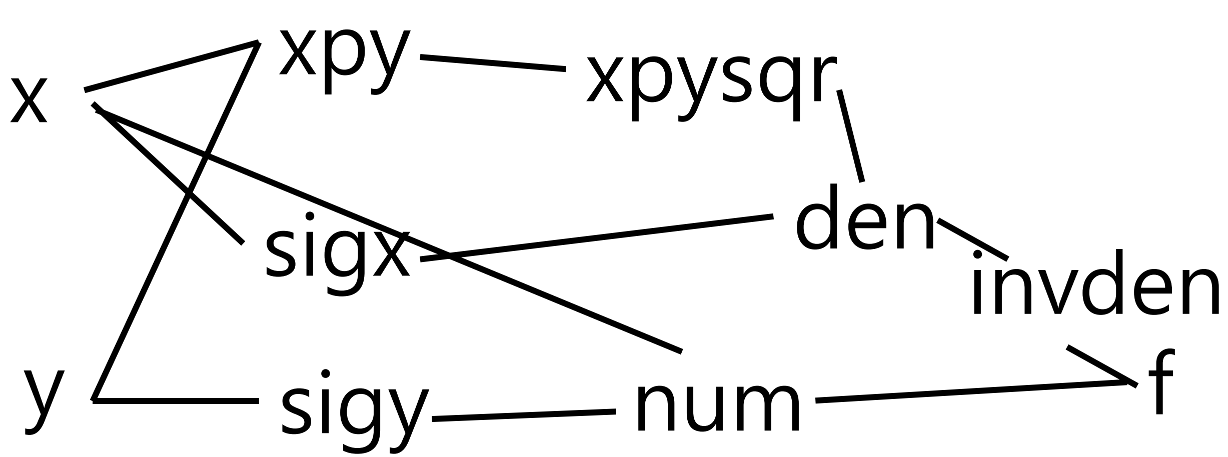

```
w = [2,-3,-3] # assume some random weights and data
x = [-1, -2]

# forward pass
dot = w[0]*x[0] + w[1]*x[1] + w[2]
f = 1.0 / (1 + math.exp(-dot)) # sigmoid function

# backward pass through the neuron (backpropagation)
ddot = (1 - f) * f # gradient on dot variable, using the sigmoid gradient derivation
dx = [w[0] * ddot, w[1] * ddot] # backprop into x
dw = [x[0] * ddot, x[1] * ddot, 1.0 * ddot] # backprop into w
# we're done! we have the gradients on the inputs to the circuit
```

역전파 다른 예시

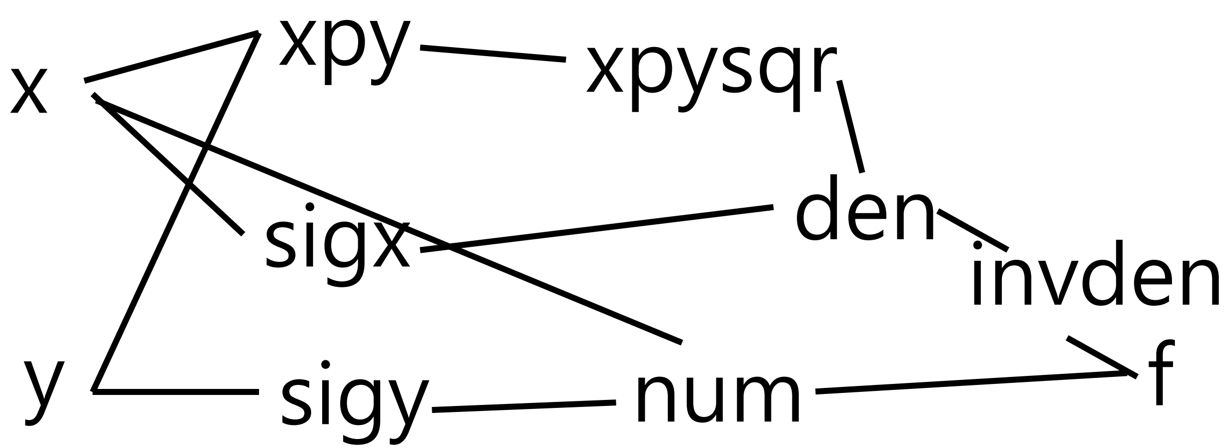
$$f(x, y) = \frac{x + \sigma(y)}{\sigma(x) + (x+y)^2}$$




```python
x = 3 # example values
y = -4

# forward pass
sigy = 1.0 / (1 + math.exp(-y)) # sigmoid in numerator   #(1)
num = x + sigy # numerator                                #(2)
sigx = 1.0 / (1 + math.exp(-x)) # sigmoid in denominator #(3)
xpy = x + y                                               #(4)
xpysqr = xpy**2                                           #(5)
den = sigx + xpysqr # denominator                         #(6)
invden = 1.0 / den                                        #(7)
f = num * invden # done!                                  #(8)
```

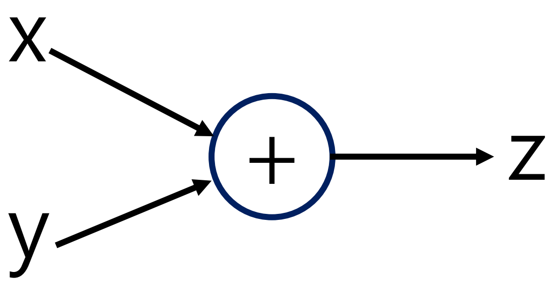
역전파 다른 예시



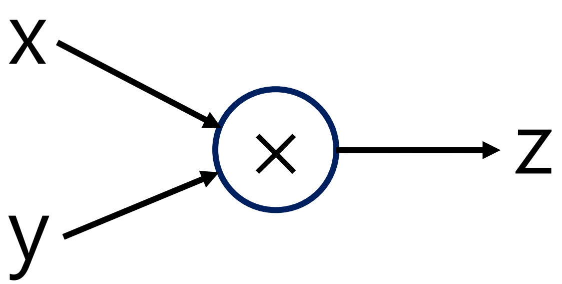

```
# backprop f = num * invden
dnum = invden # gradient on numerator                                  #(8)
dinvden = num                                                          #(8)
# backprop invden = 1.0 / den
dden = (-1.0 / (den**2)) * dinvden                                     #(7)
# backprop den = sigx + xpysqr
dsigx = (1) * dden                                                     #(6)
dxpysqr = (1) * dden                                                   #(6)
# backprop xpysqr = xpy**2
dxpy = (2 * xpy) * dxpysqr                                             #(5)
# backprop xpy = x + y
dx = (1) * dxpy                                                        #(4)
dy = (1) * dxpy                                                        #(4)
# backprop sigx = 1.0 / (1 + math.exp(-x))
dx += ((1 - sigx) * sigx) * dsigx # Notice += !! See notes below       #(3)
# backprop num = x + sigy
dx += (1) * dnum                                                       #(2)
dsigy = (1) * dnum                                                     #(2)
# backprop sigy = 1.0 / (1 + math.exp(-y))
dy += ((1 - sigy) * sigy) * dsigy                                      #(1)
# done! phew
```

역전파 패턴

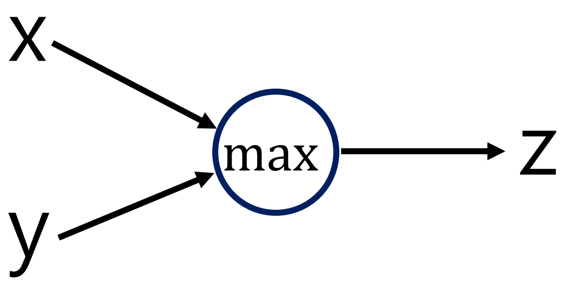
- 덧셈 게이트
  - df/dz가 df/dx, df/dy에 동일하게 전달됨



- 곱셈 게이트
  - df/dx = df/dz * y, df/dy = o
  - df/dy = df/dz * x, df/dx = o

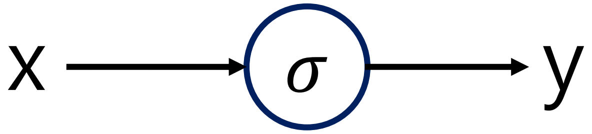


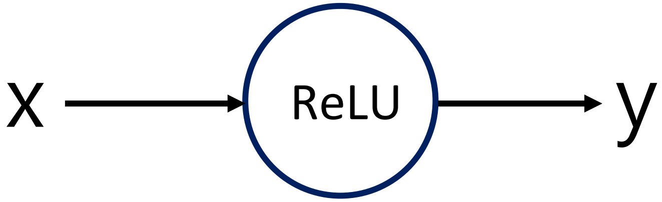
- 최대값 게이트
  - x>y이면 df/dx = df/dz, df/dy = o
  - x<y이면 df/dx = o, df/dy = df/dz



역전파 패턴

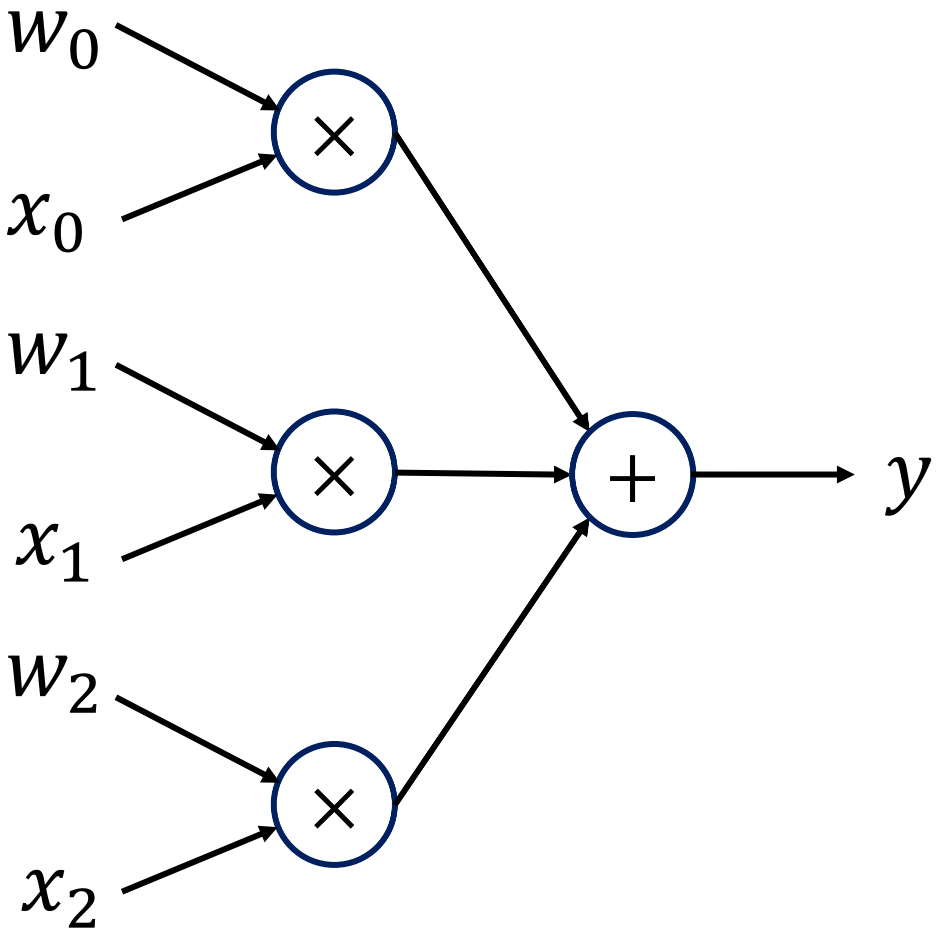
- 시그모이드 함수(σ)
  - df/dx = df/dy * (1-y)y
- ReLU 함수
  - ReLU(x) = max(0,x)
  - x>0이면 df/dx = df/dy
  - x<0이면 df/dx = 0





역전파 패턴

- 내적 형태
  - $y = w_0 x_0 + w_1 x_1 + w_2 x_2$
  - $\frac{df}{dw_0} = \frac{df}{dy} \cdot x_0$
  - $\frac{df}{dx_0} = \frac{df}{dy} \cdot w_0$
  - $\frac{df}{dw_1} = \frac{df}{dy} \cdot x_1$
  - $\dots$



역전파 패턴

- 행렬-벡터곱

$$\begin{bmatrix} w_0 & w_1 & w_2 \\ w'_0 & w'_1 & w'_2 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} y \\ y' \end{bmatrix}$$

  - $\frac{df}{dW} = \frac{df}{dy} x^T$
  - $dW = dy\, x^T$

$$\frac{\partial f}{\partial W} = \begin{bmatrix} \frac{\partial f}{\partial y} x_0 & \frac{\partial f}{\partial y} x_1 & \frac{\partial f}{\partial y} x_2 \\ \frac{\partial f}{\partial y'} x_0 & \frac{\partial f}{\partial y'} x_1 & \frac{\partial f}{\partial y'} x_2 \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial y'} \end{bmatrix} \begin{bmatrix} x_0 & x_1 & x_2 \end{bmatrix}$$

역전파 패턴

- 행렬-행렬곱
  - $Y = WX$
  - $dW = dY \cdot X^T$
  - $dX = W^T \cdot dY$

```
# forward pass
W = np.random.randn(5, 10)
X = np.random.randn(10, 3)
D = W.dot(X)

# now suppose we had the gradient on D from above in the circuit
dD = np.random.randn(*D.shape) # same shape as D
dW = dD.dot(X.T) #.T gives the transpose of the matrix
dX = W.T.dot(dD)
```