

In [1]:

```
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"
```

In [2]:

```
import numpy as np
```

## 2-1

II. 다음 함수에 대하여 변수  $x$ 와  $z$ 의 값이 모두 2 일 때 미분한 값을 구하려고 합니다 [각10점]

1.  $x$  대한 미분 값을 구하는  $\partial f / \partial x$  식을 구해보세요.  $y=2$
2.  $x$ 와  $z$ 의 값이 모두 2일 때의 미분 값을 파이썬 코드로 계산해보세요.

$$f = 2xy + 3x^2z + 4z \quad \textcircled{1} \quad \frac{\partial f}{\partial x} = 2y + 6xz$$

## 2-2

In [3]:

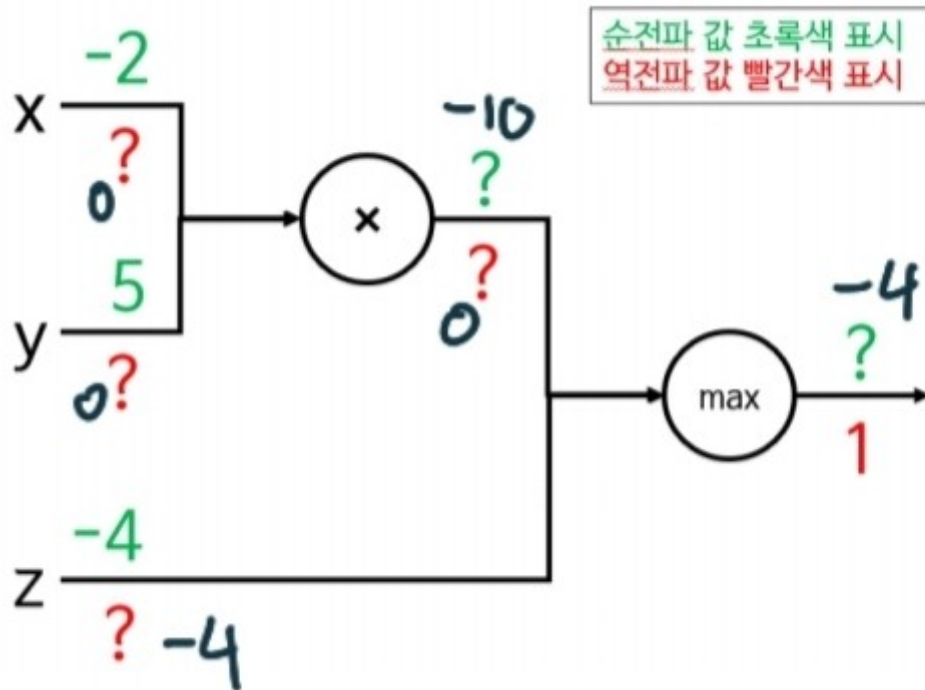
```
x, y, z = 2, 2, 2
f = 2*x*y + 3*(x**2)*z + 4*z # f 식

dfdx = 2*y + 6*x*z

print(dfdx)
```

28

## 3-1



## 3-2

In [4]:

```
x = -2; y = 5; z = -4;
p = x*y
f = max(p,z)
print(f'p = {p}, f = {f}')

dfdp = 0
dfdz = f

dpx = y
dpy = x

dfdx = dfdp * dpx
dfdy = dfdp * dpy
print(f'dfdx : {dfdx}, dfdy : {dfdy}, dfdz = {dfdz}')
```

```
p = -10, f = -4
dfdx : 0, dfdy : 0, dfdz = -4
```

## 4번

### 4-1

In [5]:

```
def sigmoid(x):
    return 1/(1+np.exp(-x))
```

In [6]:

```
input_ = np.array([0.4,0.5])
input_w = np.array([[2.0,3.0],[1,4]])
output = np.dot(input_,input_w)
final = sigmoid(output)
print(final)
```

[0.78583498 0.96083428]

**4-2**

In [7]:

```
target=np.array([1.70887704,1.4168273]) # 목표값
error = abs(target-output) # error의 절대값
w = input_w.copy()
normalize = np.array([[w[0,0]/(w[0,0]+w[1,0]),w[0,1]/(w[0,1]+w[1,1])],
                     [w[1,0]/(w[1,0]+w[0,0]),w[1,1]/(w[0,1]+w[1,1])]) # 정규화
hidden_error = np.dot(normalize,error)
print(hidden_error)
```

[1.03680156 1.15524818]

**4-3**

In [8]:

```
learning_rate=0.1
input_w -= learning_rate*np.dot(-error* output* (1-output), input_w.T)
input_w
```

