

# 201601769 나요한

## 2-1번 문제

In [1]:

```
s = {2, 3, 5, 7}
s.add(11)
s.add(13)
s.add(2)
x = len(s)
z = 9 in s
y = 2 in s
print(x)
print(y)
print(z)
print(s)
```

```
6
True
False
{2, 3, 5, 7, 11, 13}
```

## 2-2번 문제

In [2]:

```
import numpy as np

onehots = {}
onehots['cat'] = [1,0,0,0]
onehots['the'] = [0,1,0,0]
onehots['dog'] = [0,0,1,0]
onehots['sat'] = [0,0,0,1]
sentence = ['the', 'cat', 'sat']

def calc():
    list = np.array([0, 0, 0, 0])
    for i in range(len(sentence)):
        list += onehots[sentence[i]]
    return list

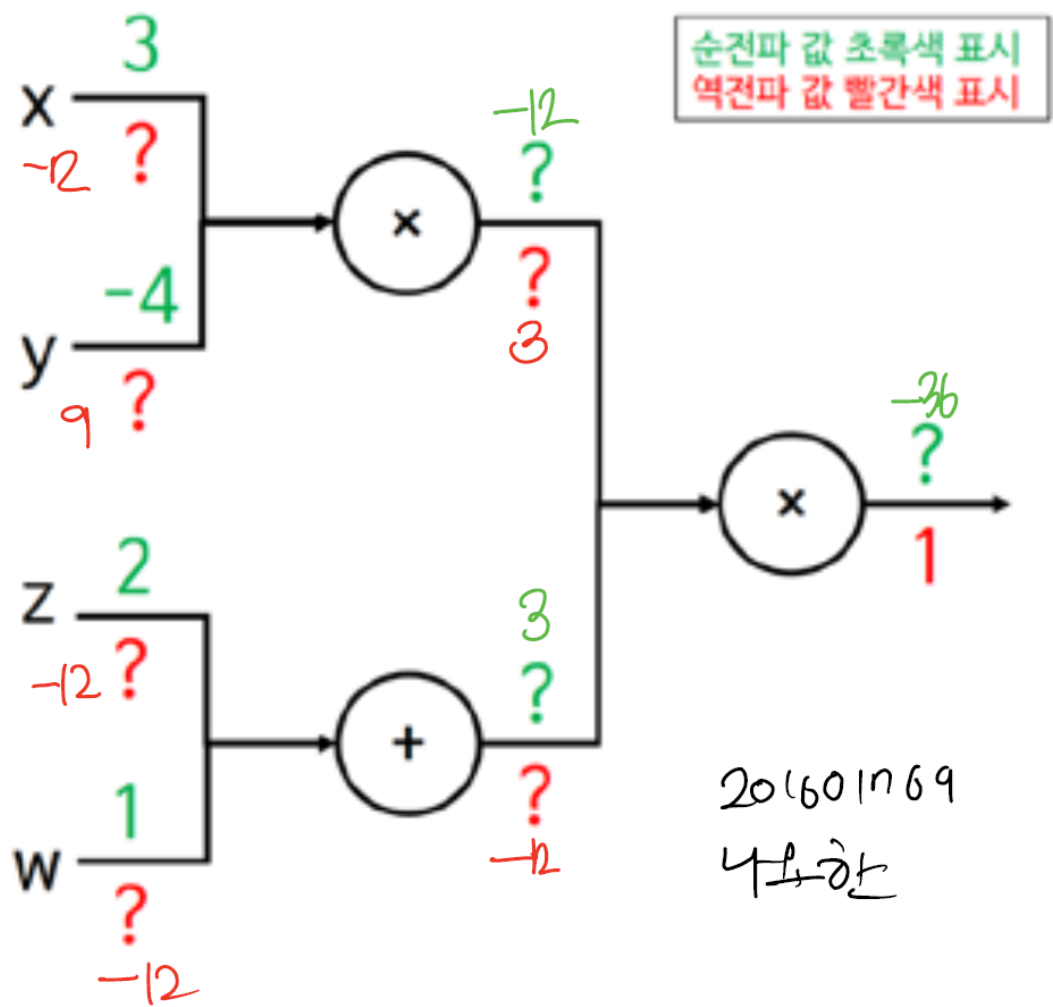
x = calc()

#array와 세트 개념으로
#넘파이의 array배열로 작성
#완전 파이썬으로 작성

print("Sentence Encoding: " + str(x))
```

Sentence Encoding: [1 1 0 1]

3-1번 문제



3-2번 문제

In [3]:

```
def mul_back_2(w1,w2,error):  
    return w2*error, w1*error  
  
def plus_back_2(error):  
    return error  
  
error = 1  
h_layer = mul_back_2(-12,3,1)  
  
x, y = mul_back_2(3, -4, h_layer[0])  
z = plus_back_2(h_layer[1])  
w = plus_back_2(h_layer[1])  
  
print(x,y,z,w)
```

-12 9 -12 -12

## 4-1번

In [4]:

```
# 랩 8.3 코드 준것.  
# 단, 시그모이드 함수 대신 2x를 넣고 미분한거 2를 넣어야함.
```

In [5]:

```
import numpy as np  
  
def activation(x):  
    return 2*x  
  
W = np.array([[0.8, 0.4], [0.9, 0.6]])  
I = np.array([[1.0], [0.5]])  
  
def forward(W, I):  
    O_k = np.dot(W, I)  
    O_k = np.array(activation(O_k))  
    return O_k  
  
O_k = forward(W,I)  
print(O_k)
```

[[2. ]  
 [2.4]]

## 4-2번

In [6]:

```

W = np.array([[0.8, 0.4], [0.9, 0.6]])
l = np.array([[1.0], [0.5]])
targets = [[1.2], [1.9]]

E_k = np.array(np.abs(targets - O_k))
#print(E_k)

def backward(W, E):
    e = np.dot(W.T, E)
    return e

def normalize(W):
    ans = np.ones([2,2])
    ans[0,0] = W[0,0]/ (W[0,1]+W[1,0])
    ans[0,1] = W[0,1]/ (W[0,1]+W[0,0])
    ans[1,0] = W[1,0]/ (W[1,0]+W[0,0])
    ans[1,1] = W[1,1]/ (W[0,1]+W[1,1])
    return ans

back = backward(normalize(W), E_k)
print(back)

```

```

[[0.75701357]
 [0.56666667]]

```

## 4-3번

In [7]:

```

lr = 0.05

W = np.array([[0.8, 0.4], [0.9, 0.6]])
l = np.array([[1.0], [0.5]])
targets = [[1.2], [1.9]]

O_k = forward(W,l)
print(O_k)
E_k = np.array(np.abs(targets - O_k))

```

```

[[2. ]
 [2.4]]

```

In [8]:

```

E_k = np.array(np.abs(targets - O_k))
Update_val = lr*(-E_k * 2*O_k * l.T)
W = W + Update_val

O_k = forward(W,l)
print(O_k)

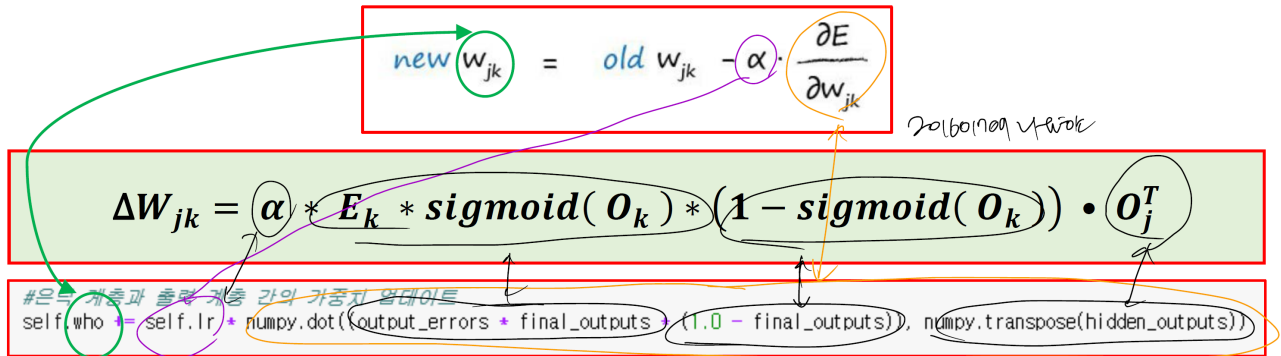
```

```

[[1.6]
 [2.1]]

```

## 4-4번



## 5-1번

인공지능은 가장 큰 개념으로 스스로 하는 모든것을 인공지능이라 하며, 머신러닝은 기계학습으로 기계적으로 선형 분류자를 만드는것을 의미하며, 딥러닝은 신경망을 활용하여 복잡한 구분자를 만드는 것을 의미한다.

## 5-2번

1950년대의 튜링테스트에서부터 음성인식, 컴퓨터 체스(딥블루), 달파철헤린지, 왓슨, 알파고로 이어지는 인공지능의 역사를 알 수 있었으며, 간단한 컴퓨터 알고리즘, 탐색알고리즘, 마르코프 결정과정, 자연어처리와 관련된 간단한 이론적인 내용을 이해할 수 있었다.

In [ ]: