

In [1]:

```
import pandas as pd
import numpy as np
```

In [2]:

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

1번

In [3]:

```
input_x = np.array([0.9,0.1,0.8])
input_w=np.array([[0.9,0.3,0.4],[0.2,0.8,0.2],[0.1,0.5,0.6]])
hidden_w = np.dot(input_w,input_x)
hidden_o = sigmoid(hidden_w)
hidden_o
```

Out[3]:

```
array([0.76133271, 0.60348325, 0.65021855])
```

In [4]:

```
hidden_w2 = np.array([[0.3,0.7,0.5],[0.6,0.5,0.2],[0.8,0.1,0.9]])
output_x = np.dot(hidden_w2,hidden_o)
output_x
```

Out[4]:

```
array([0.97594736, 0.88858496, 1.25461119])
```

In [5]:

```
result = sigmoid(output_x)
result
```

Out[5]:

```
array([0.72630335, 0.70859807, 0.77809706])
```

2번

In [6]:

```
e_output = np.array([0.8,0.5])
w_ho = np.array([[2/(2+3),3/(2+3)],[1/(1+4),4/(1+4)]])
e_hidden = np.dot(w_ho.T,e_output)
e_hidden
```

Out[6]:

```
array([0.42, 0.88])
```

In [7]:

```
w_hi = np.array([[3/(2+3),2/(2+3)],[1/(1+7),7/(1+7)]])
e_input = np.dot(w_hi.T,e_hidden)
e_input
```

Out[7]:

```
array([0.362, 0.938])
```

3번

In [8]:

```
result = np.array([0.4,0.5])
e_output = np.array([0.8,0.5])
w_output = np.array([[2.0,3.0],[1.0,4.0]])
learning_rate = 0.1
```

In [9]:

```
x=sigmoid(np.dot(w_output,result))
x1 = x*(1-x)
```

In [10]:

```
a= -learning_rate*np.dot(np.matrix(e_output*x1).T,np.matrix(result))
```

In [11]:

```
w_output-a # 그대로 w_output== ~~ 할 때 값이 끝가지 안나옴
```

Out[11]:

```
matrix([[2.00265023, 3.00331278],
        [1.0015251 , 4.00190637]])
```

4번

In [15]:

```
output = np.array([0.72630335, 0.70859807, 0.77809706]) # 1번의 output
target = np.array([0.01,0.01,0.99])
w_output = np.array([[0.3,0.7,0.5],[0.6,0.5,0.2],[0.8,0.1,0.9]])
hidden_output=np.array([0.76133271, 0.60348325, 0.65021855])
input_x = np.array([0.9,0.1,0.8])
input_w=np.array([[0.9,0.3,0.4],[0.2,0.8,0.2],[0.1,0.5,0.6]])
output_error = target - output
learning_rate=0.1
```

In [16]:

```
x=sigmoid(np.dot(w_output,hidden_output))
x1=x*(1-x)
```

In [17]:

```
w_output+=learning_rate*np.dot(np.matrix(output_error*x1).T,np.matrix(output))
```

In [18]:

```
w_output
```

Out[18]:

```
array([[0.28965805, 0.68991016, 0.48892055],
       [0.58952298, 0.48977838, 0.18877585],
       [0.80265737, 0.10259259, 0.90284687]])
```

In [19]:

```
w=w_output
```

In [20]:

```
w_output = np.array([[w[0][0]/(w[0][0]+w[1][0]+w[2][0]),w[1][0]/(w[0][0]+w[1][0]+w[2][0]),w[2][0]/(
    w[0][1]/(w[0][1]+w[1][1]+w[2][1]),w[1][1]/(w[0][1]+w[1][1]+w[2][1]),w[2][1]/(w[
    w[0][2]/(w[0][2]+w[1][2]+w[2][2]),w[1][2]/(w[0][2]+w[1][2]+w[2][2]),w[2][2]/(w[
# 가중치 설정
```

In [21]:

```
hidden_errors = np.dot(w_output.T,output_error)
```

In [22]:

```
x=sigmoid(np.dot(input_w,input_x))
x1=x*(1-x)
```

In [23]:

```
input_w+=learning_rate*np.dot(np.matrix(hidden_errors*x1).T,np.matrix(hidden_output))
```

In [24]:

```
input_w
```

Out[24]:

```
array([[0.89400046, 0.29524436, 0.39487608],
       [0.19102567, 0.79288635, 0.19233545],
       [0.09520877, 0.49620215, 0.59590803]])
```

201600779 김영민