201700949 설재혁

1. 333 신경망 Foward propagation

In [68]:

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

import numpy as np

def sigmoid_function(x):
    return 1.0 / (1 + np.exp(-x))

input_data = [0.9, 0.1, 0.8] # 입력값

W_input_hidden = [[0.9, 0.3, 0.4], [0.2, 0.8, 0.2], [0.1, 0.5, 0.6]] # 입력 계층과 은닉
W_hidden_output = [[0.3, 0.7, 0.5], [0.6, 0.5, 0.2], [0.8, 0.1, 0.9]] # 은닉 계층과 출력

X_hidden = np.dot(W_input_hidden, input_data) # 은닉 계층의 입력값 = 초기 입력값과 가중치 간의

input_data = sigmoid_function(X_hidden) # 은닉 계층의 결과값 == 출력 계층의 입력값

X_output = np.dot(W_hidden_output, input_data) # 출력 계층 입력값과 가중치 간의 행렬곱

output_data = sigmoid_function(X_output) # 출력 계층의 결과값

print('은닉 계층의 입력값: ', X_hidden)
print('은닉 계층의 결과값(=출력계층의 입력값): ', input_data)
print('출력 계층의 결과값: ', output_data)
```

```
은닉 계층의 입력값:  [1.16 0.42 0.62]
은닉 계층의 결과값(=출력계층의 입력값):  [0.76133271 0.60348325 0.65021855]
출력 계층의 결과값:  [0.72630335 0.70859807 0.77809706]
```

2. 222 신경망 Backward propagation

In [69]:

```
e_output_1 = 0.8; e_output_2 = 0.5 # 출력 계층의 두 노드의 오차
W_hidden_output= [[2.0, 3.0], [1.0, 4.0]]
W_input_hidden = [[3.0, 2.0], [1.0, 7.0]]

e_hidden_1 = e_output_1 * W_hidden_output[0][0]/np.sum(W_hidden_output[0]) + e_outpu
e_hidden_2 = e_output_2 * W_hidden_output[1][1]/np.sum(W_hidden_output[1]) + e_outpu
e_input_1 = e_hidden_1 * W_input_hidden[0][0]/np.sum(W_input_hidden[0]) + e_hidden_2
e_input_2 = e_hidden_2 * W_input_hidden[1][1]/np.sum(W_input_hidden[1]) + e_hidden_1

print('은닉 계층의 첫 번째 노드의 오차: ',e_hidden_1)
print('은닉 계층 두 번째 노드의 오차: ', e_hidden_2)
print('입력 계층의 첫 번째 노드의 오차: ',e_input_1)
print('입력 계층 두 번째 노드의 오차: ', e_input_2)
```

```
은닉 계층의 첫 번째 노드의 오차:  0.42000000000000004
은닉 계층 두 번째 노드의 오차:  0.8800000000000001
입력 계층의 첫 번째 노드의 오차:  0.3620000000000001
입력 계층 두 번째 노드의 오차:  0.9380000000000002
```

3. 222 신경망 Weight Update

In [70]:

```
e_output_1 = 0.8 # 출력계층 오차
alpha = 0.1 # 학습률
output_j = np.array([0.4, 0.5]) # 은닉계층에서의 결과 값
W_hidden_output = np.array([2.0, 3.0]) # 은닉 계층과 출력 계층 간의 가중치
sum_of_weight = np.dot(W_hidden_output, output_j) # 입력 신호의 가중치 합

result = -(e_output_1 * sigmoid_function(sum_of_weight) * (1 - sigmoid_function(sum_
variance = alpha * result # 변화량
updatedWeight = W_hidden_output[0] - variance # 업데이트된 가중치

print("오차 기울기: ", result)
print("변화량: ", variance)
print("업데이트 된 가중치(W_11): ", updatedWeight)
```

```
오차 기울기:  -0.02650226143703718
변화량:  -0.002650226143703718
업데이트 된 가중치(W_11):  2.002650226143704
```

4. 333 신경망 역전파 오차값과 가중치 업데이트

In [71]:

```
import scipy.special

targets = np.array([[0.01], [0.01], [0.99]]) # 목표 값
actuals = np.array([[0.726], [0.708], [0.778]]) # 실제 값

W_input_hidden = np.array([[0.9, 0.3, 0.4], [0.2, 0.8, 0.2], [0.1, 0.5, 0.6]]) # 입력
W_hidden_output = np.array([[0.3, 0.7, 0.5], [0.6, 0.5, 0.2], [0.8, 0.1, 0.9]]) # 은

E_output = targets - actuals # 오차
print("출력 계층 오차 \n", E_output)

W_hidden_output_sum1 = W_hidden_output.sum(axis=1, dtype="float") # 은닉 계층과 출력 계층
W_hidden_output_norm = W_hidden_output.T / W_hidden_output_sum1 # 정규화

E_hidden = np.dot(W_hidden_output_norm, E_output) # 은닉 계층 오차
print('\n은닉 계층 오차 \n', E_hidden)

W_input_hidden_sum1 = W_input_hidden.sum(axis=1,  dtype="float") # 입력 계층과 은닉 계층
W_input_hidden_norm = W_hidden_output.T / W_input_hidden_sum1 # 정규화

E_input = np.dot(W_hidden_output_norm, E_hidden) # 입력 계층 오차
print('\n입력 계층 오차 \n', E_input)

output_j = np.array([0.4, 0.5, 0.6]) # 은닉 계층에서의 결과값 (가공의 값)
sum_of_weight = np.dot(W_hidden_output, output_j) # 입력 신호의 가중치 합 (아직 이해가 잘 안
print('\n입력 신호의 가중치 합 \n', sum_of_weight)

result = -(E_output * sigmoid_function(sum_of_weight) * (1 - sigmoid_function(sum_of
print('\n오차 기울기 \n', result)
variance = alpha * result # 변화량 == 학습률 * 오차 기울기
print('\n변화량 \n', variance)

print('\n업데이트 전 가중치\n', W_hidden_output)
updatedWeight = W_hidden_output - variance # 업데이트 된 가중치
print('\n업데이트 된 가중치\n', updatedWeight)
```

```
출력 계층 오차
 [[-0.716]
 [-0.698]
 [ 0.212]]

은닉 계층 오차
 [[-0.37113162]
 [-0.59081709]
 [-0.24005128]]

입력 계층 오차
 [[-0.4536006 ]
 [-0.41376828]
 [-0.33463112]]

입력 신호의 가중치 합
 [0.77 0.61 0.91]

오차 기울기
 [[ 0.06195407  0.08166464  0.08790945]
 [ 0.06039657  0.07961162  0.08569943]
 [-0.01834394 -0.02418003 -0.02602905]]
```

변화량
```
 [[ 0.00619541  0.00816646  0.00879094]
 [ 0.00603966  0.00796116  0.00856994]
 [-0.00183439 -0.002418   -0.00260291]]
```

업데이트 전 가중치
```
 [[0.3 0.7 0.5]
 [0.6 0.5 0.2]
 [0.8 0.1 0.9]]
```

업데이트 된 가중치
```
 [[0.29380459 0.69183354 0.49120906]
 [0.59396034 0.49203884 0.19143006]
 [0.80183439 0.102418   0.90260291]]
```