

< Source code >

```
def calculate(input):
    global weights
    global bias
    activation = bias
    for i in range(2):
        activation += weights[i] * input[i]
        if activation >= 0.0:
            return 1.0
        else:
            return 0.0

def train_weights(X, y, l_rate, n_epoch):
    global weights
    global bias
    for epoch in range(n_epoch):
        sum_error = 0.0
        for row, target in zip(X, y):
            actual = calculate(row)
            error = target - actual
            bias = bias + l_rate * error
            sum_error += error**2
            for i in range(2):
                weights[i] = weights[i] + l_rate * error * row[i]
            print(weights, bias)
        print('에포크 번호=%d, 학습률=%.3f, 오류=%.3f' % (epoch, l_rate, sum_error))
        if sum_error == 0:
            break
    return weights

X = [[160, 55], [163, 43], [165, 48], [170, 80], [175, 76], [180, 70]]
y = [0, 0, 0, 1, 1, 1]

weights = [0.0, 0.0]
bias = 0.0
l_rate = 0.1
n_epoch = 100
weights = train_weights(X, y, l_rate, n_epoch)
print(weights, bias)
```

〈 질문 〉

1. 학습을 위한 데이터 세트에서 케이스에 대한 레이블은 어떤 값을 사용해야 하는가?

>> 레이블 값으로는 성별을 사용해야 한다.

2. 학습에 필요한 가중치 개수는 몇 개 인가?

>> 학습에 필요한 가중치 개수는 2개이다.

3. 학습 완료 후 분류가 완전히 되는가? 최종 가중치 조합을 쓰시오.

>> 완전히 분류가 된다.

최종 가중치 조합:

```
weights = [-18.0, 48.3000000000000004]
```

```
bias = -0.300000000000000004
```

4. 학습 종료 조건을 어떻게 설정하는 것이 좋은가? 에포크를 충분히 크게 설정해 학습하는 방법은 적절한가?

>> 학습 종료 조건은 `sum_error = 0` 일때로 설정하였다. `sum_error`를 크게 해주는 것이 좋지만 `error`가 0이 되면 더 이상 학습할 필요가 없기 때문에 실행시간만 길어지게 된다.

따라서 에포크를 너무 크게 해버리면 모델을 찾았음에도 불구하고 반복하기 때문에 낭비가 크다.

5. 7번 케이스로 173cm, 78kg의 여자가 추가될 때 학습이 잘 이뤄지는가? 잘 안되는 경우 이에 대한 설명을 추가하시오.

>> 선형으로 분류 할 수 없기 때문에 학습이 잘 되지 않는다.

7번 케이스 여자의 몸무게가 낮았다면 선형으로 분류 가능했을 것이다.