

1-layer perceptron의 구현 (n-input AND gate)

컴퓨터과학부 2018920031 유승리 | 인공지능 | 과제 #1

1. 프로그램 실행 방법

① input의 개수를 입력하여 설정한다. 예시로 input의 개수로 2를 입력하여 input이 2개인 1-layer perceptron을 구현해본다.

```
*****
***** n-input AND gate (1-layer perceptron) *****
*****
> the number of inputs : 2
```

② input의 개수로 2를 입력했으므로 RESULT 표에 input x_1 과 x_2 의 조합인 (0,0), (0,1), (1,0), (1,1)이 출력되며, 정해진 범위 내에서 랜덤으로 설정된 w_1 , w_2 , $theta$ 의 초기 값을 통해 output이 출력된다.

[초기 값 범위]

w_1, w_2	$0.0 \leq w_1, w_2, \dots, w_n < 100.0$
$theta$	$10.0 \leq theta < 50.0$

```
> the number of inputs : 2

----- RESULT -----
  x2 |  x1  || out
-----
  0  |   0  ||  0
  0  |   1  ||  1
  1  |   0  ||  1
  1  |   1  ||  1

> the number of wrong outputs : 2
> the values of variables :
(1) W1 = 64.4
(2) W2 = 72.2
(3) theta = 41.9

===== MODIFICATION =====
(1) W1
(2) W2
(3) theta
> Choose one variable to modify : 1
```

이 경우, output의 값이 차례로 0111이므로 AND gate의 output인 0001과 비교했을 때 틀린 output의 개수가 2이라는 것이 출력되며, 현재의 w_1 , w_2 , $theta$ 값 또한 소수점 아래 첫째 자리까지 출력된다.

③ 틀린 output의 개수가 0이 아니기 때문에 아직 AND gate와 일치하지 않으므로 어떤 변수를 무슨 값으로 수정할지 사용자로부터 숫자를 입력 받는다. 이때, 수정을 원하는 값은 실수 값으로도 입력이 가능하다.

```
===== MODIFICATION =====
(1) W1
(2) W2
(3) theta
> Choose one variable to modify : 3
> Enter the modified value (ex. 000.0) : 123.4
----- RESULT -----
X2 | X1 || out
-----
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1
> the number of wrong outputs : 0
> the values of variables :
(1) W1 = 64.4
(2) W2 = 72.2
(3) theta = 123.4
*****
***** We've got a correct AND gate! *****
*****
> the number of attempts : 1
계속하려면 아무 키나 누르십시오 . . .
```

이 경우, θ 값을 기존의 41.9에서 123.4로 변경하기 위해 위와 같이 숫자를 입력했다. 수정된 θ 값에 의해 output의 값이 차례로 0001이 나와 틀린 output의 개수가 0이기 때문에, 현재의 w_1 , w_2 , θ 값이 출력된 이후에 AND gate와 일치하는 output을 얻었다는 메시지가 총 시도 횟수와 함께 출력된다. 그리고 프로그램이 종료된다. 만약 값을 수정해도 틀린 output의 개수가 0이 아니라면, 그것이 0이 될 때까지 무한히 반복하여 사용자의 새로운 수정 값을 입력 받는다.

2. 프로그램을 빨리 종료하는 방법

> 기본 원리

AND gate를 구현하기 위해서는 다음 조건을 모두 따라야 한다.

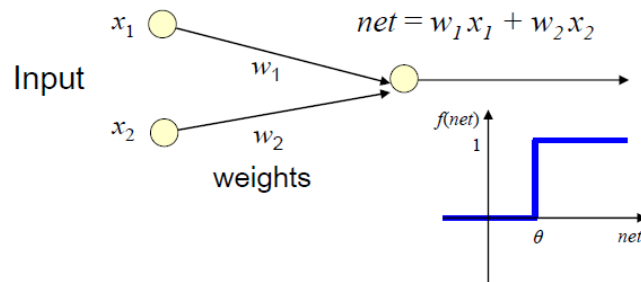


그림 출처 | 인공지능 1주차 강의자료

① 모든 input 값이 $x_1 = x_2 = \dots = x_n = 1$ 일 때만 output = 1 이어야 한다.

$$\Rightarrow w_1 x_1 + w_2 x_2 + \dots + w_n x_n = w_1 \cdot 1 + w_2 \cdot 1 + \dots + w_n \cdot 1 = w_1 + w_2 + \dots + w_n > \theta$$

② 하나 이상의 input 값이 0일 때는 output = 0 이어야 한다.

$$\Rightarrow (w_1 \sim w_n \text{ 중에서 } 0 \sim n-1 \text{ 개 값의 모든 조합의 합}) \leq \theta$$

따라서 위 두 조건을 최대한 단순한 단계를 통해 만족시킬 수 있도록 $w_1, w_2, \dots, w_n, \theta$ 값들을 사용자가 판단했을 때 제일 유리한 방법으로 수정해야한다.

> 예시

다음 프로그램 실행 결과들은 input의 개수가 2인 상황에서 프로그램 실행을 24회 반복한 결과이다. 첫 실행에서 프로그램이 종료되는 경우는 제외하였다. 또한 수정 값은 모두 연산 결과를 소수점 아래 둘째자리에서 반올림하여 소수점 아래 첫째자리까지 입력하였다.

① w 값을 수정하는 경우

- w 값들 중 가장 큰 값을 매 수정마다 2로 나누어 간다.
- 모든 output이 0이 되는 경우, w 값들 중 가장 작은 값을 1.5배 해가며 틀린 output의 개수가 0이 될 때까지 프로그램을 실행한다. (w 값에 연산하는 숫자는 임의로 정한 것이다.)

```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0

> the values of variables :
(1) W1 = 11.6
(2) W2 = 9.7
(3) theta = 14.9

***** We've got a correct AND gate! *****

> the number of attempts : 3

```

```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0

> the values of variables :
(1) W1 = 39.7
(2) W2 = 29.3
(3) theta = 45.3

***** We've got a correct AND gate! *****

> the number of attempts : 2

```

```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0

> the values of variables :
(1) W1 = 7.9
(2) W2 = 7.8
(3) theta = 10.7

***** We've got a correct AND gate! *****

> the number of attempts : 5

```

```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0

> the values of variables :
(1) W1 = 24.7
(2) W2 = 21.4
(3) theta = 28.8

***** We've got a correct AND gate! *****

> the number of attempts : 2

```

```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0

> the values of variables :
(1) W1 = 35.1
(2) W2 = 17.7
(3) theta = 49.8

***** We've got a correct AND gate! *****

> the number of attempts : 1

```

```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0

> the values of variables :
(1) W1 = 19.2
(2) W2 = 19.5
(3) theta = 33.9

***** We've got a correct AND gate! *****

> the number of attempts : 5

```

```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0

> the values of variables :
(1) W1 = 17.9
(2) W2 = 20.6
(3) theta = 26.2

***** We've got a correct AND gate! *****

> the number of attempts : 4

```

```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0

> the values of variables :
(1) W1 = 13.3
(2) W2 = 13.0
(3) theta = 24.1

***** We've got a correct AND gate! *****

> the number of attempts : 2

```

```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0

> the values of variables :
(1) W1 = 10.9
(2) W2 = 6.8
(3) theta = 12.0

***** We've got a correct AND gate! *****

> the number of attempts : 3

```

```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0

> the values of variables :
(1) W1 = 24.6
(2) W2 = 38.3
(3) theta = 46.9

***** We've got a correct AND gate! *****

> the number of attempts : 3

```

```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0

> the values of variables :
(1) W1 = 12.0
(2) W2 = 11.0
(3) theta = 18.4

***** We've got a correct AND gate! *****

> the number of attempts : 5

```

```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0

> the values of variables :
(1) W1 = 33.9
(2) W2 = 21.7
(3) theta = 34.5

***** We've got a correct AND gate! *****

> the number of attempts : 3

```

```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0
> the values of variables :
(1) W1 = 24.7
(2) W2 = 18.8
(3) theta = 38.0

***** We've got a correct AND gate! *****

> the number of attempts : 2

```

```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0
> the values of variables :
(1) W1 = 21.0
(2) W2 = 24.6
(3) theta = 30.6

***** We've got a correct AND gate! *****

> the number of attempts : 3

```

```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0
> the values of variables :
(1) W1 = 27.4
(2) W2 = 30.9
(3) theta = 43.2

***** We've got a correct AND gate! *****

> the number of attempts : 2

```

```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0
> the values of variables :
(1) W1 = 9.9
(2) W2 = 9.1
(3) theta = 14.8

***** We've got a correct AND gate! *****

> the number of attempts : 6

```

```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0
> the values of variables :
(1) W1 = 16.5
(2) W2 = 18.1
(3) theta = 23.6

***** We've got a correct AND gate! *****

> the number of attempts : 2

```

```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0
> the values of variables :
(1) W1 = 23.2
(2) W2 = 28.8
(3) theta = 37.8

***** We've got a correct AND gate! *****

> the number of attempts : 2

```

```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0
> the values of variables :
(1) W1 = 20.6
(2) W2 = 28.5
(3) theta = 43.8

***** We've got a correct AND gate! *****

> the number of attempts : 1

```

```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0
> the values of variables :
(1) W1 = 8.0
(2) W2 = 10.4
(3) theta = 10.4

***** We've got a correct AND gate! *****

> the number of attempts : 5

```

```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0
> the values of variables :
(1) W1 = 26.3
(2) W2 = 28.0
(3) theta = 30.1

***** We've got a correct AND gate! *****

> the number of attempts : 2

```

```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0
> the values of variables :
(1) W1 = 21.8
(2) W2 = 4.9
(3) theta = 26.6

***** We've got a correct AND gate! *****

> the number of attempts : 1

```

```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0
> the values of variables :
(1) W1 = 16.3
(2) W2 = 22.7
(3) theta = 30.8

***** We've got a correct AND gate! *****

> the number of attempts : 3

```

```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0
> the values of variables :
(1) W1 = 31.0
(2) W2 = 36.0
(3) theta = 39.6

***** We've got a correct AND gate! *****

> the number of attempts : 2

```

⇒ w 값을 수정하는 방법으로 프로그램을 24회 실행시킨 결과, 평균 3.625회의 시도 끝에 프로그램이 종료되었다.

② θ 값을 수정하는 경우

- 다음 조건을 만족시켜야 하므로 w 값들 중 가장 큰 값으로 θ 값을 수정한다.

$$w \text{ 값들 중 가장 큰 값} \leq \theta < w \text{ 값들의 합}$$

- input의 개수가 2인 경우, 두 w 값이 같거나 적어도 하나의 w 값이 0이 아닌 이상 1회 실행만으로 프로그램 종료 가능하다. 만약 두 w 값이 같거나 적어도 하나의 w 값이 0이라면 그 w 값을 임의의 수만큼 증가시킨 후 θ 값을 다시 위와 같이 수정하면 된다.

```
----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0
> the values of variables :
(1) W1 = 77.8
(2) W2 = 97.8
(3) theta = 97.8

***** We've got a correct AND gate! *****
> the number of attempts : 1
```

```
----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0
> the values of variables :
(1) W1 = 7.4
(2) W2 = 57.4
(3) theta = 57.4

***** We've got a correct AND gate! *****
> the number of attempts : 1
```

```
----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0
> the values of variables :
(1) W1 = 89.1
(2) W2 = 33.0
(3) theta = 89.1

***** We've got a correct AND gate! *****
> the number of attempts : 1
```

```
----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0
> the values of variables :
(1) W1 = 41.2
(2) W2 = 28.0
(3) theta = 41.2

***** We've got a correct AND gate! *****
> the number of attempts : 1
```

```
----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0
> the values of variables :
(1) W1 = 16.2
(2) W2 = 58.6
(3) theta = 58.6

***** We've got a correct AND gate! *****
> the number of attempts : 1
```

```
----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0
> the values of variables :
(1) W1 = 18.6
(2) W2 = 14.2
(3) theta = 18.6

***** We've got a correct AND gate! *****
> the number of attempts : 1
```

```
----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0
> the values of variables :
(1) W1 = 63.5
(2) W2 = 75.8
(3) theta = 75.8

***** We've got a correct AND gate! *****
> the number of attempts : 1
```

```
----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0
> the values of variables :
(1) W1 = 86.5
(2) W2 = 56.3
(3) theta = 86.5

***** We've got a correct AND gate! *****
> the number of attempts : 1
```

```
----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0
> the values of variables :
(1) W1 = 11.9
(2) W2 = 67.7
(3) theta = 67.7

***** We've got a correct AND gate! *****
> the number of attempts : 1
```



```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0
> the values of variables :
(1) W1 = 56.8
(2) W2 = 30.9
(3) theta = 56.8

***** We've got a correct AND gate! *****

> the number of attempts : 1

```

```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0
> the values of variables :
(1) W1 = 73.7
(2) W2 = 79.7
(3) theta = 79.7

***** We've got a correct AND gate! *****

> the number of attempts : 1

```

```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0
> the values of variables :
(1) W1 = 57.5
(2) W2 = 40.3
(3) theta = 57.5

***** We've got a correct AND gate! *****

> the number of attempts : 1

```

```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0
> the values of variables :
(1) W1 = 87.0
(2) W2 = 21.6
(3) theta = 87.0

***** We've got a correct AND gate! *****

> the number of attempts : 1

```

```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0
> the values of variables :
(1) W1 = 47.7
(2) W2 = 30.3
(3) theta = 47.7

***** We've got a correct AND gate! *****

> the number of attempts : 1

```

```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0
> the values of variables :
(1) W1 = 20.4
(2) W2 = 5.3
(3) theta = 20.4

***** We've got a correct AND gate! *****

> the number of attempts : 1

```

```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0
> the values of variables :
(1) W1 = 87.3
(2) W2 = 15.4
(3) theta = 87.3

***** We've got a correct AND gate! *****

> the number of attempts : 1

```

```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0
> the values of variables :
(1) W1 = 77.6
(2) W2 = 76.8
(3) theta = 77.6

***** We've got a correct AND gate! *****

> the number of attempts : 1

```

```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0
> the values of variables :
(1) W1 = 50.3
(2) W2 = 30.1
(3) theta = 50.3

***** We've got a correct AND gate! *****

> the number of attempts : 1

```

```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0
> the values of variables :
(1) W1 = 23.0
(2) W2 = 81.9
(3) theta = 81.9

***** We've got a correct AND gate! *****

> the number of attempts : 1

```

```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0
> the values of variables :
(1) W1 = 47.9
(2) W2 = 51.3
(3) theta = 51.3

***** We've got a correct AND gate! *****

> the number of attempts : 1

```

```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0
> the values of variables :
(1) W1 = 44.0
(2) W2 = 48.8
(3) theta = 48.8

***** We've got a correct AND gate! *****

> the number of attempts : 1

```

```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0

> the values of variables :

(1) W1 = 52.0
(2) W2 = 98.0
(3) theta = 98.0

***** We've got a correct AND gate! *****

> the number of attempts : 1

```

```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0

> the values of variables :

(1) W1 = 52.2
(2) W2 = 42.2
(3) theta = 52.2

***** We've got a correct AND gate! *****

> the number of attempts : 1

```

```

----- RESULT -----
X2 | X1 || out
0 | 0 || 0
0 | 1 || 0
1 | 0 || 0
1 | 1 || 1

> the number of wrong outputs : 0

> the values of variables :

(1) W1 = 16.4
(2) W2 = 79.6
(3) theta = 79.6

***** We've got a correct AND gate! *****

> the number of attempts : 1

```

⇒ *theta* 값을 수정하는 방법으로 프로그램을 24회 실행시킨 결과, 모두 1회의 시도만에 프로그램이 종료되었다.

- 하지만 input의 개수가 2보다 큰 경우부터는 *w* 값들 중 가장 큰 값으로 *theta* 값을 변경한 이후, 그 다음으로 큰 *w* 값을 *theta*에 더해가며 프로그램이 종료될 때까지 반복한다.

```

----- RESULT -----
X3 | X2 | X1 || out
0 | 0 | 0 || 0
0 | 0 | 1 || 1
0 | 1 | 0 || 1
0 | 1 | 1 || 1
1 | 0 | 0 || 1
1 | 0 | 1 || 1
1 | 1 | 0 || 1
1 | 1 | 1 || 1

> the number of wrong outputs : 6

> the values of variables :

(1) W1 = 31.1
(2) W2 = 25.5
(3) W3 = 35.3
(4) theta = 11.5

```

```

----- RESULT -----
X3 | X2 | X1 || out
0 | 0 | 0 || 0
0 | 0 | 1 || 0
0 | 1 | 0 || 0
0 | 1 | 1 || 0
1 | 0 | 0 || 0
1 | 0 | 1 || 0
1 | 1 | 0 || 0
1 | 1 | 1 || 1

> the number of wrong outputs : 0

> the values of variables :

(1) W1 = 31.1
(2) W2 = 25.5
(3) W3 = 35.3
(4) theta = 66.4

***** We've got a correct AND gate! *****

> the number of attempts : 2

```

⇒ 따라서 input이 2개일 때 가장 빨리 프로그램을 빨리 종료하는 방법은 대체적으로 위와 같이 *theta* 값을 수정하는 것이다. 또한 input이 2개보다 많을 경우에 첫 시도에서 *w* 값을 수정하는 방법을 시행한다면, 1회만에 프로그램이 끝나지 않을 때 그 다음 시도에서는 *theta* 값을 수정하는 것이 프로그램을 빨리 종료할 수 있는 방법이다.

3. 소스코드

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

/**
 * input 차원이 n인 1-layer perceptron을 구현하는 프로그램
 * 컴퓨터과학부 2018920031 유승리
 *
 * x[n]: int
 * w[n]: float (초기 값: 0.0 <= w[n] < 100.0)
 * theta: float (초기 값: 10.0 <= theta < 50.0)
 */

void calculation(int xRow, int xCol, float * w, int ** x, int * out, float theta);
void showResult(int xRow, int xCol, float * w, int ** x, int * out, float theta, int * pw);
void modifyValue(int xCol, float * w, float * pt);

int main()
{
    int input; // input의 개수 (n)
    printf("*****\n");
    printf("***** n-input AND gate (1-layer perceptron) *****\n");
    printf("*****\n");
    printf("\n\n > the number of inputs : ");
    scanf("%d", &input); // 사용자로부터 input의 개수 입력 받아 저장
    printf("\n");

    int i, j;
    int xRow = 1 << input; // == 2^n
    int xCol = input; // == n

    int ** x; // input인 x[2^n][n], int형 2차원 배열
    float * w; // weight인 w[n], float형 1차원 배열
    int * out; // output인 out[2^n], int형 1차원 배열

    int num;
    int wrong = 0; // 틀린 output의 개수
    int * pw = &wrong;
    int cnt = 0; // 맞는 output을 얻을 때까지 값을 수정한 횟수

    srand(time(NULL)); // 중복 없는 난수 생성을 위해 현재 시간을 seed로 설정
    float theta = (float)(rand() % 400) / 10 + 10; // 랜덤으로 생성되는 초기 값: 10.0 <= theta < 50.0
    float * pt = &theta;

    w = (float *)calloc(input, sizeof(float)); // w[n]을 생성하기 위한 동적 할당 (calloc)
    for (i = 0; i < input; i++)
    {
        w[i] = (float)(rand() % 1000) / 10; // 랜덤으로 생성되는 초기 값: 0.0 <= w[n] < 100.0
    }

    out = (int *)calloc(xRow, sizeof(int)); // out[2^n]을 생성하기 위한 동적 할당 (calloc)

    x = (int **)calloc(xRow, sizeof(int*)); // x[2^n][n]을 생성하기 위한 동적 할당 (calloc)
    for (i = 0; i < xRow; ++i)
    {
        x[i] = (int *)calloc(xCol, sizeof(int));
    }
}
```

```

// 10진수 -> 2진수 변환을 이용하여 x[2^n][n]에 input 값을 설정
for (i = 0; i < xRow; i++)
{
    num = i;
    for (j = 0; j < xCol; j++)
    {
        x[i][j] = num % 2;
        num = num / 2;

        if (num == 0)
            break;
    }
}

// 맞는 output을 얻을 때까지 프로그램 기능 무한 반복
while (1)
{
    calculation(xRow, xCol, w, x, out, theta);
    showResult(xRow, xCol, w, x, out, theta, pw);

    if (wrong == 0) // 각 input 별 output이 모두 맞으면 프로그램 종료, 총 시도 횟수 출력
    {
        printf("\n\n\n*****\n");
        printf("***** We've got a correct AND gate! *****\n");
        printf("*****\n");
        printf("\n > the number of attempts : %d\n\n\n", cnt);
        break;
    }
    else
        modifyValue(xCol, w, pt); // w[n], theta 값 수정

    cnt++; // 맞는 output을 얻을 때까지 값을 수정한 횟수 1 증가
    wrong = 0; // 틀린 output의 개수 초기화
}

// 동적 할당된 메모리 반환
for (i = 0; i < xRow; i++)
    free(x[i]);
free(x);
free(w);
free(out);

return 0;
}

// output = f(net)을 구하는 함수
void calculation(int xRow, int xCol, float * w, int ** x, int * out, float theta)
{
    int i, j;
    float mul, sum = 0; // sum == net

    for (i = 0; i < xRow; i++)
    {
        // net 값 연산, index가 같은 x와 w를 곱한 값들의 총합
        for (j = 0; j < xCol; j++)
        {
            mul = w[j] * x[i][j];
            sum += mul;
        }

        // net 값과 theta 값과의 비교에 따른 f(net) 연산
        if (sum > theta)
            out[i] = 1;
        else
            out[i] = 0;
    }
}

```

}

```
// 연산 결과 표, 틀린 output 개수, 현재 w[n], theta의 값을 보여주는 함수
```

$$\{$$

```
int i, j;
```

// 결과를 표 형식으로 출력

```
printf("\n----- RESULT ----- \n\n");
```

```
for (i = xCol; i > 0; i--)
```

```
printf("  X%d |", i);
```

```
printf("| out\n");
```

```
for (i = 0; i < xCol + 1; i++)
```

```
printf("-----");
```

```
printf("\n");
```

```
for (i = 0; i < xRow; ++i)
```

$$\{$$

```
for (j = xCol - 1; j >= 0; j--)
```

```
printf("  %d  |", x[i][j]);
```

```
printf("| %d\n", out[i]);
```

3

```
// out[xRow-1]을 제외한 output이 0이 아닐 때마다 틀린 output의 개수 1 증가
```

```
for (i = 0; i < xRow-1; i++)
```

 $\{$

```
if (out[i] != 0)
```

```
(*pw)++;
```

3

```
// out[xRow-1]이 1이 아닐 때 틀린 output의 개수 1 증가
```

```
if (out[xRow - 1] != 1)
```

```
(*pw)++;
```

```
printf("\n > the number of wrong outputs : %d\n", *pw); // 이번 attempt에서 틀린 output의 개수 출력
```

```
// 현재 w[n], theta의 값을 소수점 아래 첫째 자리까지 출력
```

```
printf("\n > the values of variables : \n\n");
```

```
for (i = 0; i < xCol; i++)
```

```
printf("    (%d) W%d = %.1f\n", i + 1, i + 1, w[i]);
```

```
printf("    (%d) theta = %.1f\n", i + 1, theta);
```

$$\}$$

```
// 실수 값을 사용자로부터 입력 받아 w[n], theta 값을 수정하는 함수
```

```
void modifyValue(int xCol, float * w, float * pt)
```

$$\{$$

```
int i;
```

```
int num;
```

```
float mod; // 사용자로부터 수정할 값을 입력 받는 float형 변수
```

```
printf("\n\n\n\n===== MODIFICATION =====\n\n");
```

```
for (i = 0; i < xCol; i++)
```

```
printf("    (%d) W%d\n", i + 1, i + 1);
```

```
printf("    (%d) theta\n", i + 1);
```

// w[n], theta 중 어떤 변수를 무슨 값으로 수정할지 사용자로부터 입력 받는 동작

```
printf("\n > Choose one variable to modify : ");
```

```
scanf("%d", &num);
```

```
printf("\n > Enter the modified value (ex. 000.0) : "); // float형 변수이므로 정수가 아닌 실수도 입력 가능
```

```
scanf("%f", &mod);
```

```
// 사용자가 theta 값을 수정하는 경우
```

```
if (num == i + 1)
    *pt = mod;
```

```
// 사용자가 w[n] 값을 수정하는 경우
```

```
for (i = 0; i < xCol; i++)
{
    if (num == i + 1)
        w[i] = mod;
}
```

```
// 사용자가 실수로 num 값으로 범위에서 벗어난 값을 입력한 경우 다시 반복
```

```
if (num < 1 || num > xCol + 1)
{
    printf("\n\n ** Out of range! Please enter a number among the indexes above (");
    for (i = 1; i <= xCol; i++)
        printf("%d, ", i);
    printf("%d). **\n", i);
    modifyValue(xCol, w, pt);
}
```

```
}
```