

파이썬(Python)으로 구현하는

Classification

– simple • multi-variable example –

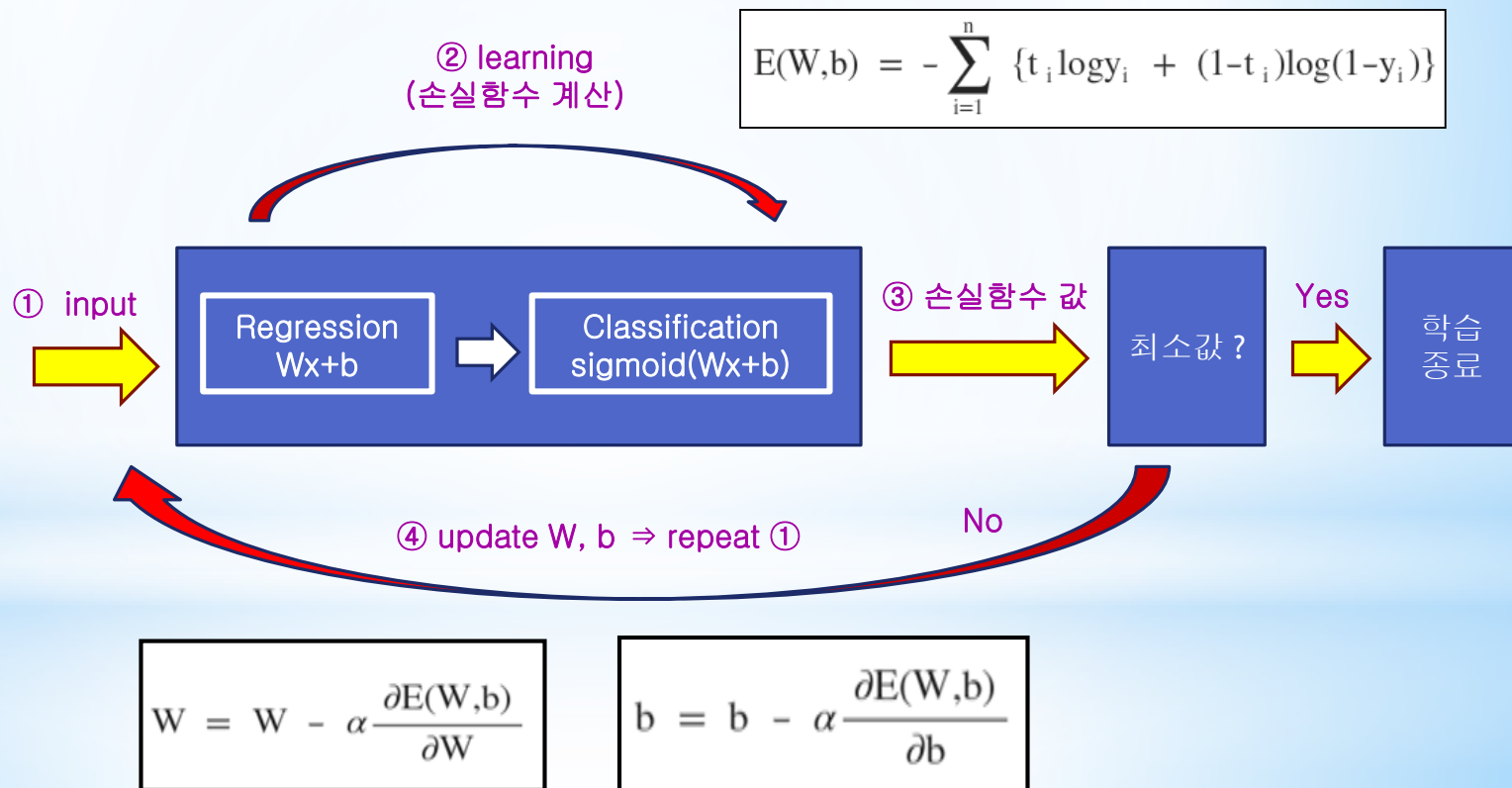
예제 1

[1] 먼저 loss_func, error_val, predict, sigmoid 함수를 구현하고, numerical_derivative 함수를 이용하여 다음의 분류 시스템을 구현하시오

[2] class 버전으로 구현하시오

공부시간 (x)	Fail/Pass (t)
2	0
4	0
6	0
8	0
10	0
12	0
14	1
16	1
18	1
20	1

training data



예제 2-1

[1] 학습데이터(Training Data) 준비

예습시간 ($x1$)	복습시간 ($x2$)	Fail/Pass (t)
2	4	0
4	11	0
6	6	0
8	5	0
10	7	1
12	16	1
14	8	1
16	3	1
18	7	1



xdata, tdata 구현

[2] 가중치 W , 바이어스 b 초기화

W , b 구현

[3] 분류를 위한 손실함수 $E(W,b)$ 및 예측 함수 $\text{predict}(\dots)$ 구현

$\text{loss_func}(\dots)$ / $\text{predict}(\dots)$ 구현

[5] 학습율 (learning rate) 초기화 및 손실함수가 최소가 될 때까지 W , b 업데이트

수치미분을 이용하여
 W , b 업데이트 코드 구현

예제 2-1

[6] 미래 값 예측

```
test_data = np.array([3, 17]) # (예측, 복측) = (3, 17) => Fail (0)  
predict(test_data)
```

```
(array([0.12867978]), 0)
```

```
test_data = np.array([5, 8]) # (예측, 복측) = (5, 8) => Fail (0)  
predict(test_data)
```

```
(array([0.00099032]), 0)
```

```
test_data = np.array([7, 21]) # (예측, 복측) = (7, 21) => Pass (1)  
predict(test_data)
```

```
(array([0.99998955]), 1)
```

```
test_data = np.array([12, 0]) # (예측, 복측) = (12, 0) => Pass (1)  
predict(test_data)
```

```
(array([0.63499634]), 1)
```

Multi-variable 분류코드를
Classification 클래스 버전으로 구현

예제 2-3

[1] 다음의 데이터를 만족하는 분류 시스템을 구현하시오

```
xdata = np.array([ [0, 0], [0, 1], [1, 0], [1, 1] ])
```

```
tdata = np.array([0, 0, 0, 1])
```

```
test_data = np.array([ [0, 0], [0, 1], [1, 0], [1, 1] ])
```

[2] 다음의 데이터를 만족하는 분류 시스템을 구현하시오

```
xdata = np.array([ [0, 0], [0, 1], [1, 0], [1, 1] ])
```

```
tdata = np.array([0, 1, 1, 0])
```

```
test_data = np.array([ [0, 0], [0, 1], [1, 0], [1, 1] ])
```