

머신러닝 / 딥러닝

Linear Regression

– NotBatch / Class 예제 –

Not-Batch 예제

[조건] 기존 함수 `loss_func`, `error_val`, `predict`, `numerical_derivative` 등을 어떤 부분도 수정하지 말고 현재 상태 그대로 사용하여야 함.

Training Data 가 아래와 같을 때, 즉 matrix를 만들기 위해 `reshape` 을 사용하지 않고 `vector` 그대로 사용하여야 할 경우 다음의 코드를 어떻게 변경해야 하는지 기술하시오

```
import numpy as np

x_data = np.array([1, 2, 3, 4, 5])
t_data = np.array([2, 3, 4, 5, 6])

print("x_data.shape = ", x_data.shape, ", t_data.shape = ", t_data.shape)

x_data.shape = (5,) , t_data.shape = (5,)
```

```
learning_rate = 1e-2 # 발산하는 경우, 1e-3 ~ 1e-6 등으로 바꾸어서 실행

f = lambda x : loss_func(x_data, t_data)

print("Initial error value = ", error_val(x_data, t_data), "Initial W = ", W, "\n", ", b = ", b )

for step in range(8001):

    W -= learning_rate * numerical_derivative(f, W)

    b -= learning_rate * numerical_derivative(f, b)

    if (step % 400 == 0):
        print("step = ", step, "error value = ", error_val(x_data, t_data), "W = ", W, ", b = ", b )
```

class 예제

[조건] 다음과 같은 LinearRegressionTest class 를 구현하시오

[1] Training Data 는 다음과 같이 np.loadtxt 를 이용하여 읽어들이야 한다

```
loaded_data = np.loadtxt('./data-01.csv', delimiter=',', dtype=np.float32)

x_data = loaded_data[:, 0:-1]
t_data = loaded_data[:, [-1]]

# 데이터 자원 및 shape 확인
print("x_data.ndim = ", x_data.ndim, ", x_data.shape = ", x_data.shape)
print("t_data.ndim = ", t_data.ndim, ", t_data.shape = ", t_data.shape)

x_data.ndim = 2 , x_data.shape = (25, 3)
t_data.ndim = 2 , t_data.shape = (25, 1)
```

[2] 객체 생성시에 입력파라미터는 입력데이터, 정답데이터, 학습률, 반복횟수 이다.

즉, obj = LinearRegressionTest(xdata, tdata, learning_rate, iteration_count) 형식으로 생성되어야 한다

```
# LinearRegressionTest 객체를 만들기 위해 4개의 파라미터 필요
# 1st : 입력데이터, 2nd : 정답데이터
# 3rd : learning rate, 4th : iteration count
obj = LinearRegressionTest(x_data, t_data, 1e-5, 10001)

obj.train()
```

[3] 객체가 생성되었다면 train 메서드를 통해서 학습을 완료 하여야 한다.

즉, obj.train() 을 통해서 loss function 이 최소가 되는 최적의 가중치 W, 바이어스 b 를 계산하여야 한다

```
test_data = np.array([100, 98, 81])

obj.predict(test_data)

array([179.57413757])
```