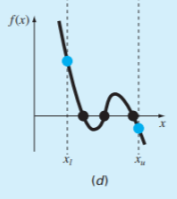
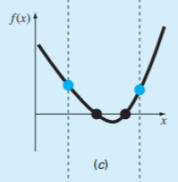
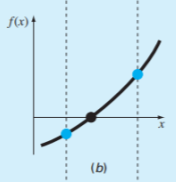
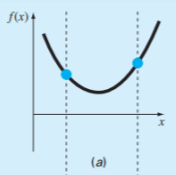
**학 과 : 소프트웨어학과 학 번 : 20171622 이 름 : 박건후**

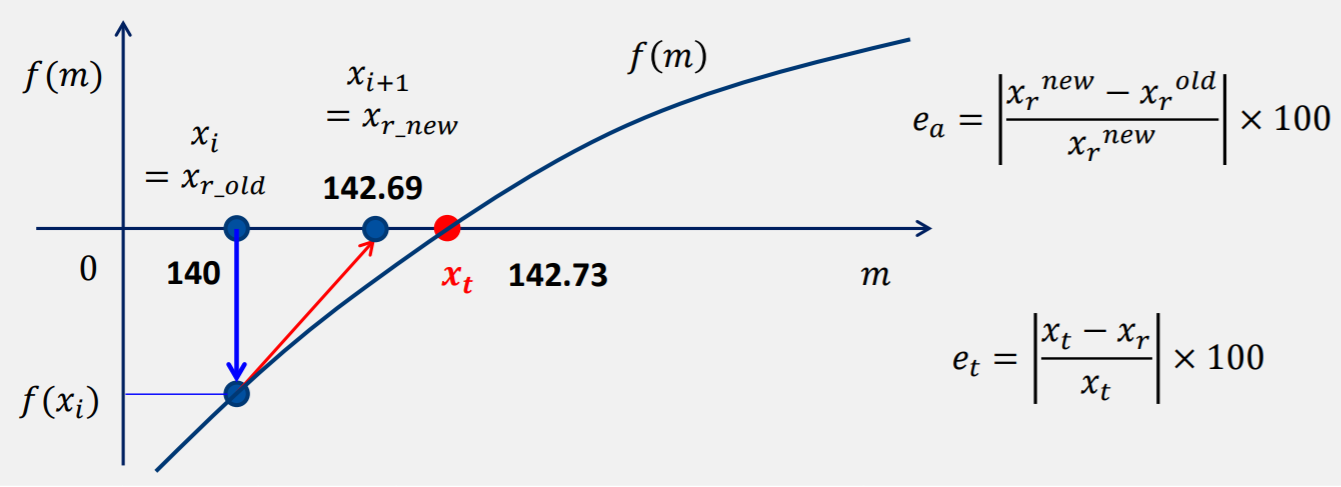
|  |  |
| --- | --- |
| 문번 | 문 제 |
| 1  (하) | 1. 아래 소스코드는 근사해를 구하는 대표적인 알고리즘 중 하나입니다. np.sign이 의미하는 바는 무엇이며, 위 알고리즘에서 < if np.sign(f[k]) != np.sign(f[k+1]): > 가 작성된 이유를 자유롭게 서술해보아라. |
| 2  (상) | 2. 아래의 빈칸 1,2에 들어갈 코드를 적으시오. |
| 3  (중) | 3. 건후는 인스타그램 날마다 인스타그램에 게시물을 업로드한다. 게시물의 ‘좋아요’수를 선형 회귀 하고 싶어 다음 코드와 그래프는 선형 회귀를 나타낸 코드다. x는 게시한 날짜로부터 지금까지의 기간이다. y는 좋아요 수다. 예를 들어 1일 전 올린 게시물의 좋아요의 수는 311개다. 다음 주어진 질문이 맞으면 T, 틀리면 F를 빈칸에 기입하라.       1. 위 코드에서 a1은 y1의 기울기, a0은 y1의 y절편을 의미한다. (T/F) 2. 회귀한 그래프와 가장 많은 오차를 나타내는 지점은 x가 2일 때다. (T/F) 3. 선형회귀한 그래프에 따르면 9일 전 게시물의 좋아요의 수는 8일 전보다 높을 것으로 예상된다. (T/F) |

1. 정답 : np.sign은 부호(sign) 판별 함수입니다. 1 (positive), 0(zero), -1(negative) 로 값을 반환합니다. if np.sign(f[k]) != np.sign(f[k+1]): 이 사용된 이유는 x1<x2, f(x1)\*f(x2)<0 일 때 근이 한 개 있거나, 근이 세 개 있음을 파악할 수 있기 때문에 사용했습니다. fp는 단조증가 함수이기 때문에 한 구간에 있어서 부호가 달라진다면 근이 한 개 있음을 방증하기에 위와 같은(<>) 코드를 사용했습니다.



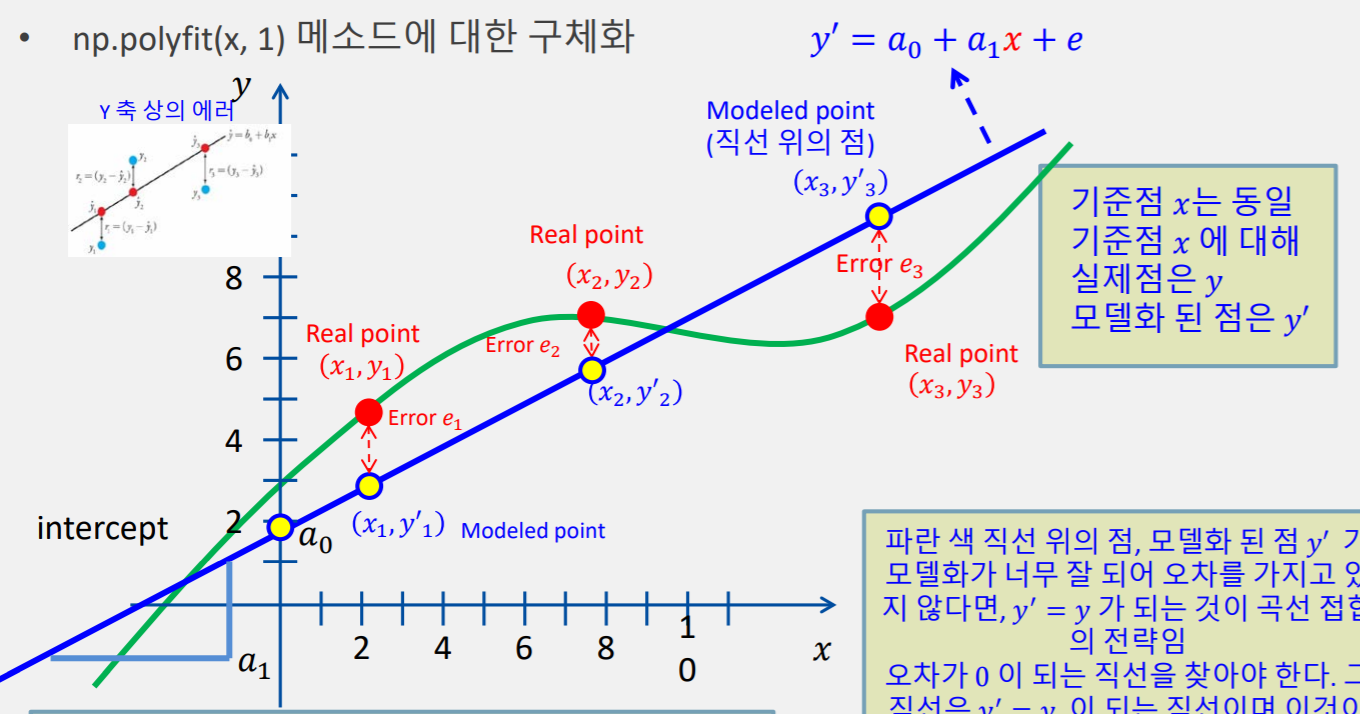
정답설명 : 𝑓(𝑥1) 과 𝑓(x𝑢) 의 부호가 다르면, (b)와 (d) 처럼 근이 한 개 있거나, 홀수 개(3개)의 근이 있음을 파악할 수 있다. 따라서 fp 그래프의 특징을 보면 단조 증가 함수기 때문에 𝑓(𝑥1) 과 𝑓(x𝑢) 의 부호가 다르면 그 구간 사이에 근이 있음을 파악할 수 있다. (난이도 하)

1. 정답 : [1] xr-func(xr)/dfunc(xr) [2] np.float(xr)-np.float(xrold))/np.float(xr)

정답설명 : Newton-Raphson 방법은 미분법으로 접선의 기울기를 구하고, 직선을 사용하는 가위치법보다 빠르게 근사해에 접근하는 방법이다. xr의 값을 계속 새로 구함으로써 상대오차가 사용자가 정한 오차값에 근접하도록 하는 것이 Newton-Raphson 방법이다. 새로운 xr (xr\_new)은 이전 xr (oldxr)에서 접선을 그어 x축과 만나는 값이다. 따라서 xr-func(xr)/dfunc(xr) 이 [1]에 들어갈 코드다. 상대오차는 {|(이론값)-(실험값)}/(이론값)×100으로 표현할 수 있다. ea는 상대오차를 구하는 코드이기 때문에 np.float(xr)-np.float(xrold))/np.float(xr) 로 표현하는 것이 옳다. (난이도 상)

3. 정답 : (1) T / (2) T / (3) T / (4) T

정답설명 :



1. a0과 a1 의 값은 편미분하여 손실함수의 최소값을 구하는 과정 중 나온 것이다. 다시 말해 미분(순간변화율, 기울기값)이 0이 되도록 하는𝑎0, 𝑎1값을 찾기 위해 고전 지식 공학방법인 편미분을 사용했다. 따라서 a0과 a1을 설정할 때 각각 y절편, 기울기로 설정하였다. 따라서 T다.
2. 선형회귀한 결과로 나타낸 그래프와 실제값이 차이가 많이 나는 지점은 x가 1 또는 4일 때다. 따라서 F다.
3. 선형회귀한 결과로 나타난 그래프로 보았을 때 감소하고 있는 그래프를 나타내고 있으므로 9일 전 게시물의 좋아요 수는 8일 전 게시물보다 낮을 것으로 예상된다. 따라서 F다.