

202122852 이혜준 AI소프트웨어학과 (1번)

문제: HW

$$\min_{x_1, x_2, x_3 \in \mathbb{R}} f(x_1, x_2, x_3) = (x_1 - 4)^2 + x_3^2 x_1 + (x_2 + 1)^2 + 6$$

경사하강법을 이용하여 위의 문제를 풀어라.

코드:

```
library(Deriv)

f <- function(x1, x2, x3)
  (x1 - 4)^2 + x3^2 * x1 + (x2 + 1)^2 + 6

x <- c(0,0,0)
alpha <- 0.1

f_prime_x1 <- Deriv(f, "x1")
f_prime_x2 <- Deriv(f, "x2")
f_prime_x3 <- Deriv(f, "x3")

for (i in 1:100) {
  grad <- c(
    f_prime_x1(x[1], x[2], x[3]),
    f_prime_x2(x[1], x[2], x[3]),
    f_prime_x3(x[1], x[2], x[3])
  )

  x <- x - alpha * grad
  cat("x:", x, "f(x):", f(x[1], x[2], x[3]), "\n")
  Sys.sleep(0.05)
}
```

결과:

```
x: 3.999999 -0.9999997 0 f(x): 6
x: 3.999999 -0.9999997 0 f(x): 6
x: 3.999999 -0.9999998 0 f(x): 6
x: 3.999999 -0.9999998 0 f(x): 6
x: 3.999999 -0.9999999 0 f(x): 6
x: 4 -0.9999999 0 f(x): 6
x: 4 -0.9999999 0 f(x): 6
x: 4 -0.9999999 0 f(x): 6
x: 4 -0.9999999 0 f(x): 6
x: 4 -1 0 f(x): 6
x: 4 -1 0 f(x): 6
x: 4 -1 0 f(x): 6
x: 4 -1 0 f(x): 6
x: 4 -1 0 f(x): 6
x: 4 -1 0 f(x): 6
x: 4 -1 0 f(x): 6
```

경사하강법을 이용하여 문제에 주어진 함수의 최솟값을 구하였습니다.

초기값을 (0, 0, 0)으로 설정하고, 학습률을 0.1로 하여 100회 반복한 결과,

변수 x는 (4, -1, 0)에 수렴하였으며 이때 함수값 f(x)는 최솟값인 6을 가집니다.

해당 지점은 함수의 전역 최소점에 해당하며, 반복 이후 더 이상의 변화 없이 안정적인 상태에 도달하였습니다.

(2번)

문제:

HW

```
state <- as.data.frame(state.x77)
```

	Population	Income	Illiteracy	Life Exp	Murder	HS Grad	Frost	Area
Alabama	3615	3624	2.1	69.05	15.1	41.3	20	50708
Alaska	365	6315	1.5	69.31	11.3	66.7	152	566432
Arizona	2212	4530	1.8	70.55	7.8	58.1	15	113417
Arkansas	2110	3378	1.9	70.66	10.1	39.9	65	51945
California	21198	5114	1.1	71.71	10.3	62.6	20	156361
Colorado	2541	4884	0.7	72.06	6.8	63.9	166	103766
Connecticut	3100	5348	1.1	72.48	3.1	56.0	139	4862
Delaware	579	4809	0.9	70.06	6.2	54.6	103	1982
Florida	8277	4815	1.3	70.66	10.7	52.6	11	54090
Georgia	4931	4091	2.0	68.54	13.9	40.6	60	58073
Hawaii	868	4963	1.9	71.60	6.2	61.9	0	6425
Idaho	813	4119	0.6	71.87	5.3	59.3	126	82677
Illinois	11197	5107	0.9	70.14	10.3	52.6	127	55748
Indiana	5313	4458	0.7	70.88	7.1	52.9	122	36092
Iowa	2861	4628	0.5	72.56	2.3	59.0	140	55941
Kansas	2280	4669	0.6	72.58	4.5	59.9	134	81787
Kentucky	3387	3712	1.6	70.10	10.6	38.5	95	39650
Louisiana	3806	3545	2.8	68.76	13.2	42.2	32	44930
Maine	1058	3694	0.7	70.39	2.7	54.7	161	30920
Maryland	4122	5299	0.9	70.22	8.5	52.3	101	9891
Massachusetts	5814	4755	1.1	71.83	3.3	58.5	103	7826
Michigan	9111	4751	0.9	70.63	11.1	52.8	125	56617
Minnesota	3921	4675	0.6	72.96	2.3	57.6	180	79289
Mississippi	2341	3098	2.4	68.09	12.5	41.0	50	47296

Population, Income, Illiteracy, Life Exp, Frost 를 입력변수로 하고, Murder 를 출력변수로 하여 경사 하강법을 이용해 다중회귀분석을 실시하라.

코드: `library(Deviv)`

```
#데이터 불러오기
state <- as.data.frame(state.x77)

# 데이터 분포, 단위 수 확인
head(state)
summary(state)

# 데이터 정규화
state_scaled <- as.data.frame(scale(state))
head(state_scaled)
summary(state_scaled)

sum(is.na(state_scaled)) # 결측치가 있는지 확인

# 독립 변수 설정
X <- cbind(1, state_scaled$Population, state_scaled$Income, state_scaled$Illiteracy, state_scaled$Life.Exp, state_scaled$Frost)

# 종속 변수 설정
y <- state_scaled$Murder

beta <- rep(0.01, ncol(X)) # 0.0 0.0 0.0 0.0
alpha <- 0.001

# 함수 정의
f <- function(beta, x) {
  return(x %*% beta)
}

# 미분 계산
f_prime <- function(beta, x, y) {
  y_pred <- f(beta, x)
  error <- y_pred - y
  grad <- t(x) %*% error / length(y)
  return(grad)
}

# 경사하강법 학습
for (i in 1:2000) {
  grad <- f_prime(beta, X, y)
  beta <- beta - alpha * grad

  # 모든 에포크마다 출력
  y_pred <- f(beta, X)
  loss <- mean((y_pred - y)^2)
  cat(sprintf("Epoch %d: Loss = %.6f | Beta = %s\n", i, loss, paste(round(beta, 4), collapse = ", ")))
}
```

```

결과: > state <- as.data.frame(state.x77)
> # 데이터 분포, 단위 수 확인
> head(state)
      Population Income Illiteracy Life Exp Murder HS Grad Frost Area
Alabama      3615   3624         2.1   69.05   15.1   41.3    20 50708
Alaska       365   6315         1.5   69.31   11.3   66.7   152 566432
Arizona     2212   4530         1.8   70.55    7.8   58.1    15 113417
Arkansas     2110   3378         1.9   70.66   10.1   39.9    65  51945
California  21198   5114         1.1   71.71   10.3   62.6    20 156361
Colorado     2541   4884         0.7   72.06    6.8   63.9   166 103766

> summary(state)
      Population      Income      Illiteracy      Life Exp      Murder
Min.   : 365      Min.   :3098      Min.   :0.500      Min.   :67.96      Min.   : 1.400
1st Qu.: 1080      1st Qu.:3993      1st Qu.:0.625      1st Qu.:70.12      1st Qu.: 4.350
Median : 2838      Median :4519      Median :0.950      Median :70.67      Median : 6.850
Mean   : 4246      Mean   :4436      Mean   :1.170      Mean   :70.88      Mean   : 7.378
3rd Qu.: 4968      3rd Qu.:4814      3rd Qu.:1.575      3rd Qu.:71.89      3rd Qu.:10.675
Max.   :21198      Max.   :6315      Max.   :2.800      Max.   :73.60      Max.   :15.100

      HS Grad      Frost      Area
Min.   :37.80      Min.   : 0.00      Min.   : 1049
1st Qu.:48.05      1st Qu.: 66.25      1st Qu.: 36985
Median :53.25      Median :114.50      Median : 54277
Mean   :53.11      Mean   :104.46      Mean   : 70736
3rd Qu.:59.15      3rd Qu.:139.75      3rd Qu.: 81163
Max.   :67.30      Max.   :188.00      Max.   :566432

```

정규화 전 데이터

```

> # 데이터 정규화
> state_scaled <- as.data.frame(scale(state))
> head(state_scaled)
      Population      Income      Illiteracy      Life Exp      Murder      HS Grad      Frost
Alabama -0.1414316 -1.3211387   1.525758 -1.3621937   2.0918101 -1.4619293 -1.6248292
Alaska  -0.8693980  3.0582456   0.541398 -1.1685098   1.0624293  1.6828035  0.9145676
Arizona -0.4556891  0.1533029   1.033578 -0.2447866   0.1143154  0.6180514 -1.7210185
Arkansas -0.4785360 -1.7214837   1.197638 -0.1628435   0.7373617 -1.6352611 -0.7591257
California 3.7969790  1.1037155  -0.114842  0.6193415   0.7915396  1.1751891 -1.6248292
Colorado -0.3819965  0.7294092  -0.771082  0.8800698  -0.1565742  1.3361400  1.1838976

      Area
Alabama -0.2347183
Alaska   5.8093497
Arizona  0.5002047
Arkansas -0.2202212
California 1.0034903
Colorado  0.3870991

```

정규화 후 데이터 독립변수에서 제외되는 HS Grad, Area를 제외하고 보더라도 단위수나 분포 차이가 줄어든 것을 볼 수 있습니다.

```

Epoch 9989: Loss = 0.424707 | Beta = 0, 0.2654, 0.0047, 0.6599, -0.0147
Epoch 9990: Loss = 0.424706 | Beta = 0, 0.2654, 0.0047, 0.6599, -0.0147
Epoch 9991: Loss = 0.424706 | Beta = 0, 0.2654, 0.0047, 0.6599, -0.0147
Epoch 9992: Loss = 0.424706 | Beta = 0, 0.2654, 0.0047, 0.6599, -0.0147
Epoch 9993: Loss = 0.424706 | Beta = 0, 0.2654, 0.0047, 0.6599, -0.0147
Epoch 9994: Loss = 0.424706 | Beta = 0, 0.2654, 0.0047, 0.6599, -0.0147
Epoch 9995: Loss = 0.424705 | Beta = 0, 0.2654, 0.0047, 0.6599, -0.0147
Epoch 9996: Loss = 0.424705 | Beta = 0, 0.2654, 0.0047, 0.6599, -0.0147
Epoch 9997: Loss = 0.424705 | Beta = 0, 0.2654, 0.0047, 0.6599, -0.0147
Epoch 9998: Loss = 0.424705 | Beta = 0, 0.2654, 0.0047, 0.6599, -0.0147
Epoch 9999: Loss = 0.424705 | Beta = 0, 0.2654, 0.0047, 0.6599, -0.0147
Epoch 10000: Loss = 0.424705 | Beta = 0, 0.2654, 0.0047, 0.66, -0.0147

```

정규화 이후 반복은 10000회, 초기값은 (0,0,0,0,0,0), alpha(learning rate)은 0.001로 실행했을 시 각 변수들의 회귀 계수를 확인해볼 수 있으며 반복 수가 진행될수록 손실함수의 loss 값이나 각변수들의 회귀계수들이 변동이 없는 것으로 값이 수렴하여 안정적인 상태에 도달하였습니다.