Gradient Descent

Bhuvan Kapoor [23MDT1003]

2024-03-19

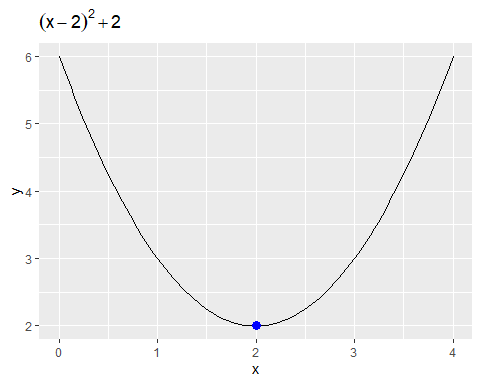
# Experiment: Gradient Descent  
  
library(grid)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(scales)  
library(ggplot2)  
library(Deriv)  
  
Formula <- function(x) {  
 (x-2)\*\*2+2  
}  
  
# visualization for the formula  
ggplot(data.frame(x=c(0,4)), aes(x)) +  
 stat\_function(fun = Formula) +  
 geom\_point(data = data.frame(x = 2, y = Formula(2)), aes(x,y),  
 color = "blue", size = 3) +  
 ggtitle(expression((x-2)\*\*2+2))



# finding the derivative for the Formula  
derivative <- Deriv(Formula,"x")  
derivative

## function (x)   
## 2 \* (x - 2)

# defining the alpha value (learning rate)  
learning\_rate <- 0.2  
  
# initializing the values  
xold <- 2.2  
xnew <- xold - learning\_rate\*derivative(xold)  
xnew

## [1] 2.12

# define the epsilon value & maximum iterations allowed  
epsilon <- 0.01  
step <- 2  
iter <- 10  
  
# records the value of x & y and add the initial guess  
xtrace <- list()  
ytrace <- list()  
xtrace[[1]] <- xold  
ytrace[[1]] <- Formula(xold)  
xtrace[[2]] <- xnew  
ytrace[[2]] <- Formula(xnew)  
  
delta <- abs(xold - xnew)  
while (delta >= epsilon & step <= iter) {  
 step <- step + 1  
 xtrace[[step]] <- xtrace[[step-1]] - learning\_rate\*derivative(xtrace[[step-1]])  
 ytrace[[step]] <- Formula(xtrace[[step]])  
 delta <- abs(xtrace[[step-1]] - xtrace[[step]])   
}  
  
record <- data.frame(x = do.call(rbind, xtrace), y = do.call(rbind, ytrace))  
record

## x y  
## 1 2.200000 2.040000  
## 2 2.120000 2.014400  
## 3 2.072000 2.005184  
## 4 2.043200 2.001866  
## 5 2.025920 2.000672  
## 6 2.015552 2.000242  
## 7 2.009331 2.000087