Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

CSE3506 Essentials of Data Analytics

Name : **Sparsh Raj**
Reg. No. : **19BPS1028**

Lab Exercise: 9: Gradient Descent Optimization

Objective: To perform Gradient Descent Optimization

Question:

Gradient Descent - Optimization

```
rm(list=ls())
```

Create a sequence of elements in a Vector to generate sequences when plotting the axes of figures or simulating data.

```
xs <- seq(0,4,len=20)

xs
```

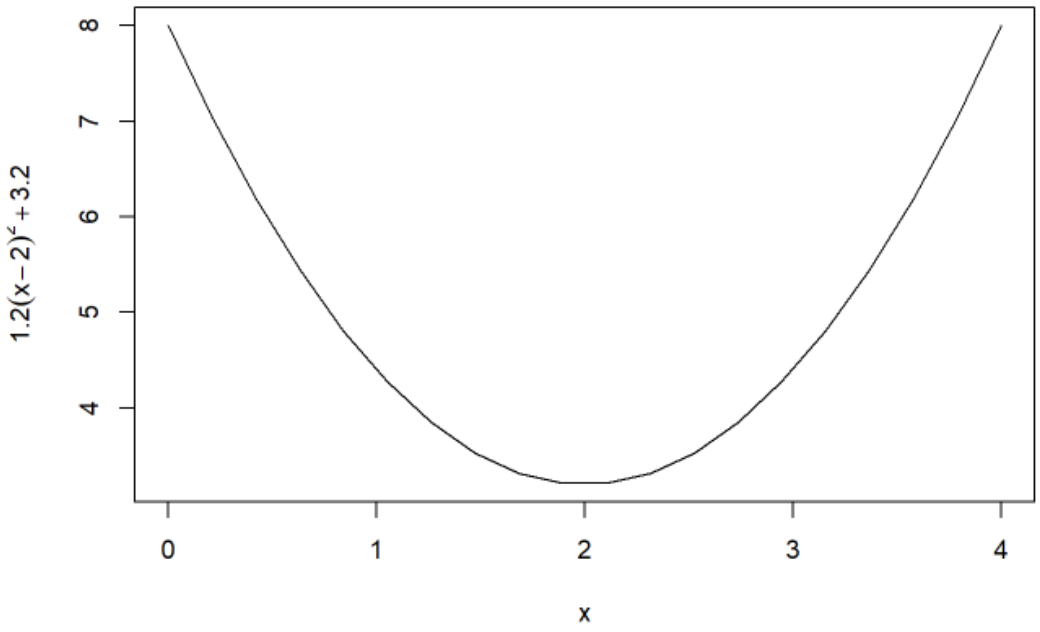
```
##  [1] 0.0000000 0.2105263 0.4210526 0.6315789 0.8421053 1.0526316 1.26315
79

##  [8] 1.4736842 1.6842105 1.8947368 2.1052632 2.3157895 2.5263158 2.73684
21

## [15] 2.9473684 3.1578947 3.3684211 3.5789474 3.7894737 4.0000000
```

Define the function we want to optimize

```
f <-  function(x) {1.2 * (x-2)^2 + 3.2}
```

Plot the function

```
plot(xs , f (xs), type="l",xlab="x",ylab=expression(1.2(x-2)^2 +3.2))
```



calculate the gradient df/dx

```
grad <- function(x){
  1.2*2*(x-2)
}
```

df/dx = 2.4(x-2), if x = 2 then 2.4(2-2) = 0

The actual solution we will approximate with gradient descent

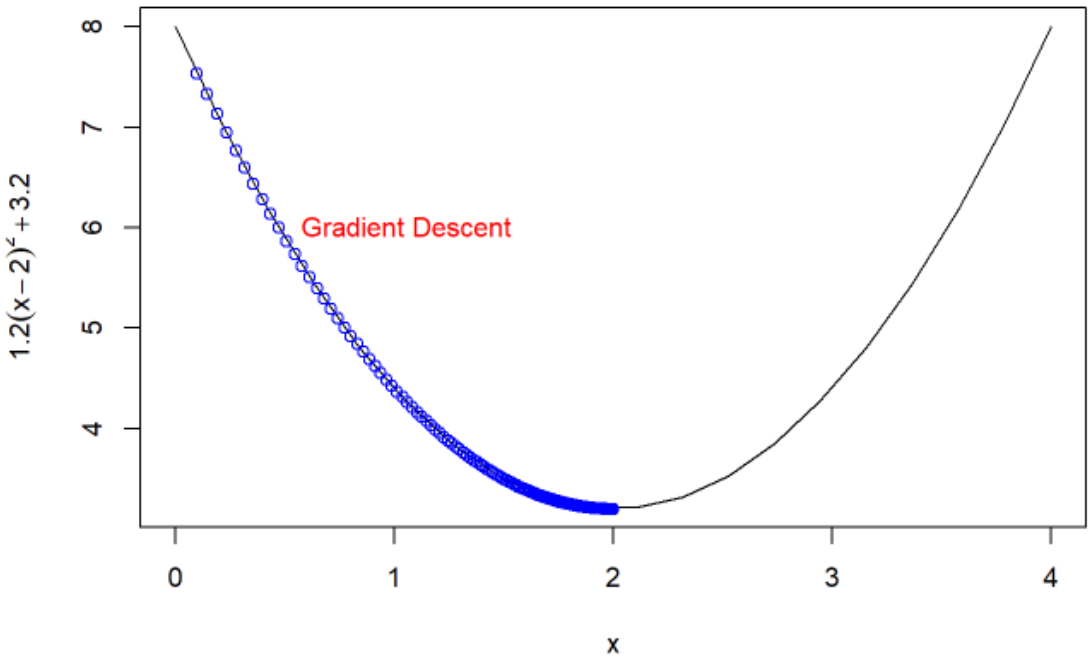
is x = 2 as depicted in the plot below

#lines (c (2,2), c (3,8), col="red",lty=2) #text (2.1,7, "Closedform solution",col="red",pos=4)

gradient descent implementation

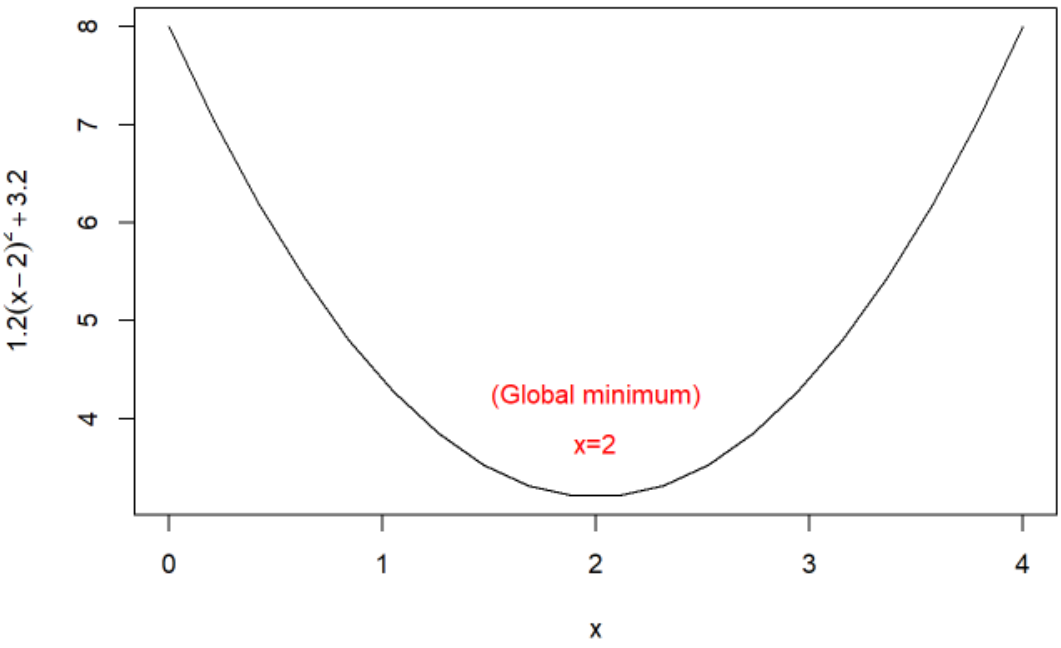
```
x <- 0.1 # initialize the first guess for x-value

xtrace <- x # store x -values for graphing purposes (initial)

ftrace <- f(x) # store y-values (function evaluated at x) for graphing purp
oses (initial)

stepFactor <- 0.01 # learning rate 'alpha'
```

```
for (step in 1:5000) {
  x <- x - stepFactor*grad(x) # gradient descent update
  xtrace <- c(xtrace,x) # update for graph
  ftrace <- c(ftrace,f(x)) # update for graph
}
plot(xs , f (xs), type="l",xlab="x",ylab=expression(1.2(x-2)^2 +3.2))
lines ( xtrace , ftrace , type="b",col="blue") # type=b (both points & lines)
text (0.5,6, "Gradient Descent",col="red",pos= 4)
```



print final value of x

```
print(x) # x converges to 2.0
## [1] 2
plot(xs , f (xs), type="l",xlab="x",ylab=expression(1.2(x-2)^2 +3.2))
text(2,4,"x=2",col="red",pos=1)
text(2,4,"(Global minimum)",col="red",pos=3)
```


1.2(x−2)²+3.2
8
7
6
5
4
(Global minimum)
x=2
0
1
2
3
4
x