VIT®

Vellore Institute of Technology

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

ABHINAV VIJAYAKUMAR

19BCE1311

CSE3506 – ESSENTIALS OF DATA ANALYTICS
LAB-10

DR. LAKSHMI PATHI JAKKAMPUTI (L21 + L22)

Tasks for Week-10: Gradient Descent with Momentum Optimizer

Aim: Apply multiple regression on mtcars dataset using momentum optimized gradient descent.

Algorithm:

1. Select columns for multiple regression

2. Give learn rate, gamma(momentum) and max iterations in function

3. Pick values for m1, m2 & c.

4. Initialize values for nu_m1, nu_m2 and nu_c to be 0.

5. Initialize iteration=0

6. If iteration < max_iteration

- **a.** Calculate y_pred
- **b.** Calculate loss function
- **c.** Update nu_m1, nu_m2 and nu_c using the below formula:
  - **i.** Nu_m1 = gamma *nu_m1 + alpha * sum((y_pred-y) * x1)
  - **ii.** Nu_m2 = gamma *nu_m2 + alpha * sum((y_pred-y) * x2)
  - **iii.** Nu_c = gamma * nu_c+alpha * sum(y_pred-y)
- **d.** Update m1, m2, c and Lf
- **e.** Print intercept, slope and loss function

7. Repeat step 5 continuously.

8. Use lm function to check for linear model.

STATISTICS:

1. Values using Momentum Optimizer:

FIELD	VALUE
C	37.2272414172067
M1	-3.87782187933926
M2	-0.0317729604979703

```
> mgd(data$wt,data$hp,data$mpg,-0.2,-0.2,32,0.000002,0.98,50000)
[1] "Optimal intercept: 37.2272414172067 Optimal slope: -3.87782187933926 -0.0317729604979703
 Loss function 97.5238773718236"
```

2. Values using lm function for Multilinear Regression:

FIELD	VALUE
C	37.22727
M1	-0.03177
M2	-3.87783

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 37.22727    1.59879  23.285  < 2e-16 ***
data$hp     -0.03177    0.00903  -3.519  0.00145 **
data$wt     -3.87783    0.63273  -6.129 1.12e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.593 on 29 degrees of freedom
Multiple R-squared:  0.8268,    Adjusted R-squared:  0.8148
F-statistic: 69.21 on 2 and 29 DF,  p-value: 9.109e-12
```

INFERENCES:

1. In momentum gradient descent loss function is not important but in gradient descent loss function is important as it is used for convergence.
2. If we put gamma as 0 the model behaves like gradient descent.
3. We can decrease the learning rate or increase the number of iterations to increase the accuracy.

Program:

```
#Momentum based Gradient Descent

mgd=function(x1,x2,y,m1,m2,c,alpha,gamma,iter){

  iterations=0

  #Lf=0

  nu_m1=0

  nu_m2=0
```

```
  nu_c=0
  while(iterations<=iter){
    y_pred=m1*x1+m2*x2+c
    Lf_new=0.5*sum((y_pred-y)^2)
    nu_m1=gamma*nu_m1+alpha*sum((y_pred-y)*x1)
    nu_m2=gamma*nu_m2+alpha*sum((y_pred-y)*x2)
    nu_c=gamma*nu_c+alpha*sum(y_pred-y)
    m1=m1-nu_m1
    m2=m2-nu_m2

    c=c-nu_c
    Lf=Lf_new
    iterations=iterations+1
  }
  paste("Optimal intercept:",c,"Optimal slope:",m1,m2,"Loss function",Lf)
}
data=mtcars
mgd(data$wt,data$hp,data$mpg,-0.2,-0.2,32,0.000002,0.98,50000)
model=lm(data$mpg~data$hp+data$wt)
summary(model)
```