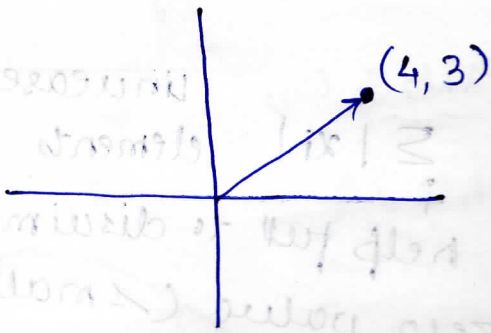
NORMS | DISTANCE METRICS

let us say we have 2 variables (x_1, x_2).

i.e $x(x_1, x_2)$

obsn	x_1	x_2
1	4	3
2		
⋮		

Thus 1st observation can be viewed as a vector with elements $\{4, 3\}$.



Thus this vector becomes 2 D. vector with x & Y coordinates.

length of this vector is the magnitude of the vector. This is called as NORM.

NORM:-

This NORM is used in Data Science to summarize or compress the information about a data vector to a single number. Thus each observation can be represented as a single number.

It is represented as $||x||$.

For the above example, we have used following formula to find the magnitude.

$$\|x\| = \sqrt{4^2 + 3^2}$$

$$= \sqrt{x_1^2 + x_2^2}$$

$$= (x_1^2 + x_2^2)^{1/2}$$

in general

$$\|x\| = \left(\sum_{i=1}^{N} |x_i|^2 \right)^{1/2}$$

This is called as Euclidean d(Norm) i.e Euclidean distance (from origin).

This is also called as L_2 Norm.

In general; 'LP. Norm.

$$\boxed{\|x\|_p = \left(\sum_{i=1}^{N} |x_i|^p \right)^{1/p}}$$

$$= \sqrt[p]{|x_1|^p + |x_2|^p + \cdots + |x_N|^p}$$

Thus

when $p = 1$ L_1 Norm $\because p = 1$

$$\|x\|_1 = \sum_{i=1}^{N} |x_i|$$

$$\|x\|_1 = |x_1| + |x_2| + \cdots + |x_n|$$

Ex: Find L_1 norm for $x = (1, -1, 2)$

$\Rightarrow$

$$\|x\|_1 = \sum_{i=1}^{3} |x_i|$$

$$= |1| + |-1| + 2$$

$$\|x\|_1 = 4$$

AAM

when $p = 2$ L_2 Norm

$$\|x\|_2 = \sqrt{\sum_{i=1}^{N} |x_i|^2}$$

$$= \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$$

Ex:- find L_2 norm for $x = (1, -1, 2)$

$$\Rightarrow \|x\|_2 = \sqrt{1^2 + (-1)^2 + (2)^2}$$

$$= \sqrt{6} = 2.449$$

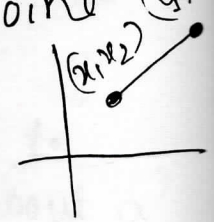
L_2 Norm < L_1 norm

Use of Norm in Data science:-

① L_1 Norm :- {Manhatten distance}

We have taken Norm as the magnitude of a vector. It is the distance between the required point & the origin.

Now let us say instead of origin we have another vector i.e another point- $Y(Y_1, Y_2)$. Then applying L_1 norm.



$$\therefore d(x, Y) = \|x - Y\|_1 = \sum_{i=1}^{N} |x_i - y_i|$$

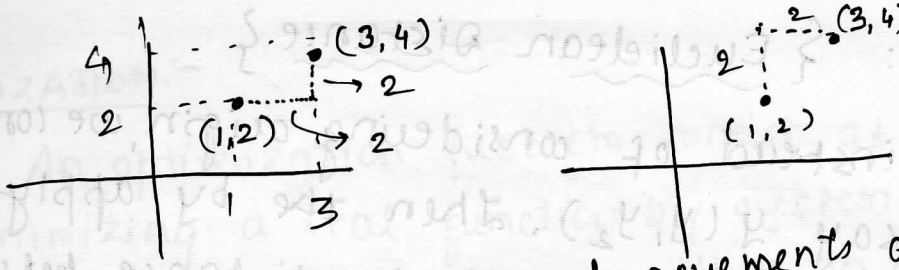
Ex:- if $x(1,2)$ & $Y(3,4)$ are two data points then

$$d(x, Y) = |1 - 3| + |2 - 4|$$

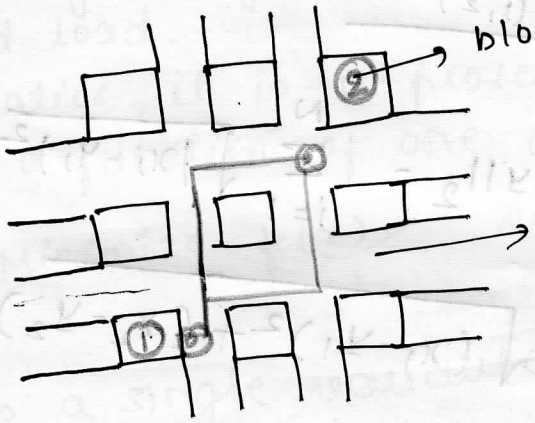
$$= 4.$$

This distance which is calculated by L_1 norm is called as Manhatten or city block distance or Taxicab distance.

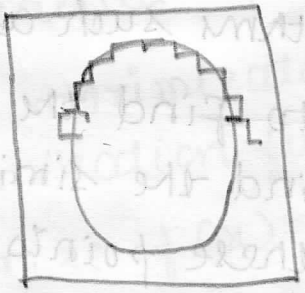
AAM



Thus only horizontal & vertical movements allowed.
There is a Manhatten ~~city~~ in A suburb in New York which is having construction in the form of blocks (grid) (showppt for geometric view of Manhatten distance.



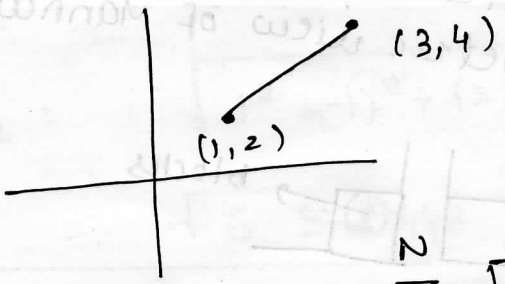
In Machine learning especially in computer vision where images are involved, this Manhatten distance is used. Image is made of pixels which are arranged or stored in the form of matrix. In general when we see any oval boundary, it is actually made up as shown below. Now when we want to go for shape matching etc we have to find the perimeter. In such cases we use Manhatten distance.



AAM

(2) L_2 Norm: {Euclidean Distance}

If instead of considering origin, we consider any pt say $y(y_1, y_2)$. Then by applying L_2 Norm we can find the distance between these two vectors x & y, which is called as Euclidean distance.



$$d(x,y) = \|x-y\|_2 = \sum_{i=1}^{N} \sqrt{(x_i - y_i)^2}$$

$$= \sqrt{(x_1-y_1)^2 + (x_2-y_2)^2}$$

Ex:- if $x(1,2)$ & $y(3,4)$ are two data points then

$$d(x,y) = \sqrt{(1-3)^2 + (2-4)^2}$$

$$= \sqrt{8}.$$

In Machine learning algorithms such as classification or clustering we need to find the distance between two points to find the similarities & dissimilarities between these points. In such cases we use Euclidean distance to find this distance. e.g KNN. K-nearest neighbour.

Note:- Euclidean distance is the shortest distance ie less than Manhatten distance. & we need to

OPTIMIZATION:-

An optimization problem consist of maximizing or minimizing a real function by systematically choosing input values from within an allowed set computing the value of the function. It involves in giving the best solution.

e.g: choosing stocks which gives maximum returns.
designing bridge that carry maximum possible/ desired load.

In mathematics, it is a process of maximizing or minimizing a function of one or more variables.

Maximize | Minimize $f(x)$ → decision variable
→ objective function

Here, x is a single variable (univariate optimization) if x is a vector (multivariate optimization).

If we have full control on x without any restriction, it called as uncontrained optimization.

If we have some restriction on x, it is called as constrained optimization.

All the maximization problems can be talled as minimization function. i.e

Maximize $f(x) \Rightarrow$ Minimize $-f(x)$.

e.g: A carpenter makes book cases in two sizes, large & small. It takes 6 hours to make a large bookcase & 2 hours to make a small one. The profit on a large bookcase is $50 & the profit on a small bookcase is $20. The carpenter can spend only 24 hours per week.

AAM

making bookcases & must make at least 2 of each size per week. Your job as a data scientist is to help your carpenter maximise his revenue.

Now, your initial thought can be the large bookcase will be most profitable.

But the "constraint" is at least 2 of each size per week.

Therefore you can suggest 2 small size (min^m required)

You have total 24 hrs.

∴ 2 small size = 4 hrs.

You are left with 20 hrs. Therefore 3 large size

3 large size = 18 hrs

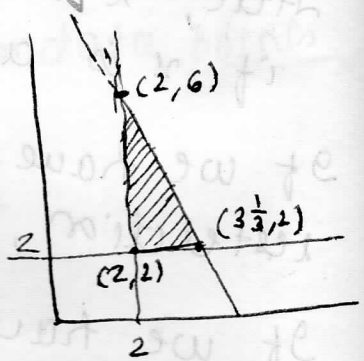
or 3 small size = 6

3 large size = 18

e.g: IPL fantasy league.

i.e Revenue :-

$3 \times 20 + 3 \times 50 = \$210$



our objective function is

$P = 50x + 20y$

if we get ask carpenter for 2 large bookcase & 6 small bookcases.

constraints are
$x \geq 2$; $y \geq 2$; $6x + 2y \leq 24$
large small hrs.

then revenue is $50(2) + 20(6)$

$= \$220$.

corner-point principle of linear programming: The maximum & minimum values of obj-fn each occur at one of the vertices of the feasible region

This is the optimum value with constraint.

It is constraint optimization. If there are no constraint like min^m of at least 2 then it becomes unconstrained optimization.

AAM

Why optimization for Machine Learning?

Almost all ML algorithms can be viewed as solutions to optimization problems.

For any optimization problem there are following components:

(i) objective function (minimization problem)
(ii) Decision variables
(iii) constraints.

If $f(x)$ is linear and all constraints are also linear, variables are continuous then it is linear programming problem.

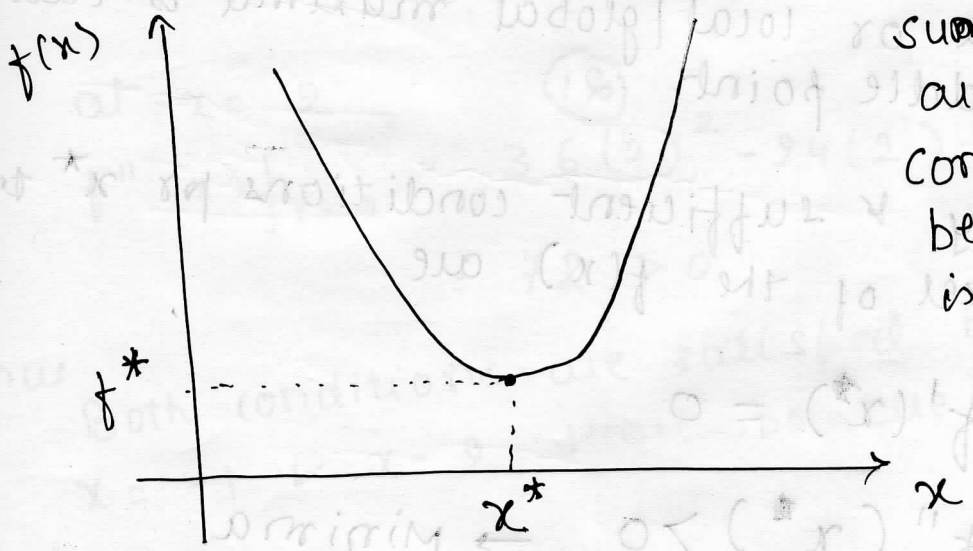
If $f(x)$ or constraint are nonlinear then called as nonlinear programming problem.

Nonlinear optimization unconstrained univariate.

Problem is $\min_x f(x)$ $x \in R$ — x is continuous any value from real nos.

univariate :- only x is the decision variable

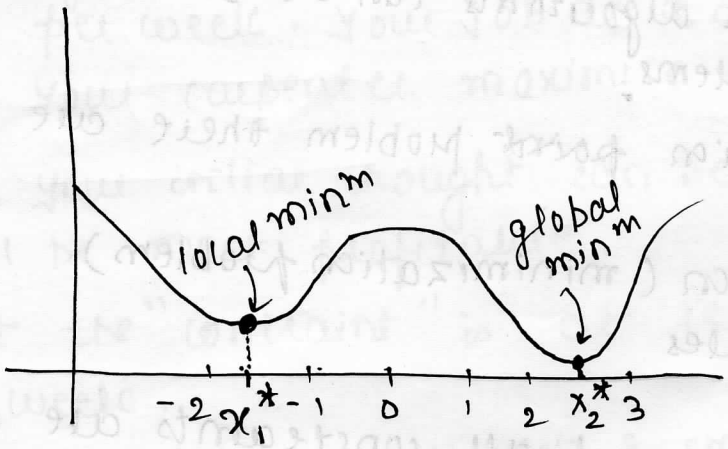
let the nonlinear function $f(x)$ is given as follows.



such functions are called as convex functions because there is only one minimum value here.

AAM

The nonlinear function can be following formats



such functions are called as non convex functions because there are multiple minima.

Job of the optimizer is to find the best solution from these local optimal solutions.

Local minimum → In vicinity to this point we cannot find a point better than x_1^*.

But if you go far away, there would be another local minimum. But this minimum is also the minimum point in the entire region hence called as Global minima.

And a point which is not local/global minima or local/global maxima is called as saddle point.

Necessary & sufficient conditions for x^* to be minimizer of the $f(x)$ are

$$f'(x^*) = 0$$

&

$$f''(x^*) > 0 \rightarrow \text{Minima}$$
$$< 0 \rightarrow \text{Maxima}$$
$$= 0 \rightarrow \text{saddle point.}$$

AAM

Ex:- $f(x) = 3x^4 - 4x^3 - 12x^2 + 3$. find optimal solun (10)

$\min_x f(x)$

Solu:- For minimal value

(i) $f'(x) = 0$

$f'(x) = 12x^3 - 12x^2 - 24x = 0$

$x^3 - x^2 - 2x = 0$

$x(x^2 - x - 2) = 0$

$x(x-2)(x+1) = 0$

$\Rightarrow$ $\boxed{x = 0, -1, 2.}$ → possibilities

(ii) $f''(x) = 36x^2 - 24x - 24$

$\therefore$ at $x = 0$

$f''(x)|_{x=0} = -24 < 0$

at $x = -1$

$f''(x)|_{x=-1} = 36(-1)^2 - 24(-1) - 24$

$= 36 > 0$

at $x = 2$

$f''(x)|_{x=2} = 36(2)^2 - 24(2) - 24$

$= 72 > 0$

Thus Both conditions are satisfied by values $x = -1$ & $x = 2$. Hence both are local minima

But at $x = -1$

$f(-1) = 3(-1)^4 - 4(-1)^3 - 12(-1)^2 + 3$

$\boxed{f(-1) = -2.}$ → at $x = -1$ (10) AAM

at $x = 2$

$f(2) = 3(2)^4 - 4(2)^3 - 12(2)^2 + 3$

$\boxed{f(2) = -29}$. at $x = 2$

Thus out of $x = -1$ & $x = 2$, we get minimum $f(x)$ at $x = 2$

Hence $x^* = 2$ is the global minima.

& $x^* = -1$ is the local minima.

HW

Minimize $f(x) = x^2 + 3x + 2$.

Non linear unconstrained multivariate optimization:-

let us take minimization problem with n variables.

Minimize $f(x)$; $x = \{x_1, x_2 \cdots x_n\}$

Here we are taking $x = \{x_1, x_2 \cdots x_n\}$ n variables.

In univariate optimization we had a condition $f'(x) = 0$ let $z = f(x)$ $\frac{\partial z}{\partial x} = f'(x) = 0$

In multivariate it is denoted as 'gradient' ∇f & is given as

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix}$$

ie $\boxed{\nabla f = 0}$ 1st condn

similarly for univariate $f''(x) > 0$.

an equivalent basis for multivariate optimization with n-variables we use $\nabla^2 f$. given by $n \times n$ Hessian matrix.

This Hessian matrix is given by

$$H(\bar{x}) = \nabla^2 f = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & & & \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

condition for multivariate x^* to be minima.

$$\nabla f(\bar{x}^*) = 0$$

&

$$\nabla^2 f(\bar{x}^*) \rightarrow \text{positive definite}$$

ie

Hessian matrix

$H(x^*)$ →

- positive definite i.e all eigen values are +ve (>0) then x^* is minima.
- negative definite i.e all eigen values are -ve (<0) then x^* is maxima
- indefinite ie both +ve & -ve values then x^* is saddle point
- semidefinite (+ve $\lambda \geq 0$) or -ve $\lambda \leq 0$) then no conclusion

Ex:- Find minimum $\bar{x}^*$ for $\Rightarrow f(x) = x_1 + 2x_2 + 4x_1^2 - x_1x_2 + 2x_2^2$.

Soln:- finding $\nabla f = 0$

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 1 + 8x_1 - x_2 \\ 2 - x_1 + 4x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} 8x_1 - x_2 + 1 \\ -x_1 + 4x_2 + 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} x_1^* \\ x_2^* \end{bmatrix} = \begin{bmatrix} -0.1935 \\ -0.548 \end{bmatrix}$$

Second condn:-

$$\nabla^2 f = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{bmatrix} = \begin{bmatrix} 8 & -1 \\ -1 & 4 \end{bmatrix}$$

This is Hessian matrix. Now finding the eigen value of Hessian matrix.

$$\begin{bmatrix} 8-\lambda & -1 \\ -1 & 4-\lambda \end{bmatrix} = 32 - 12\lambda + \lambda^2 - 1$$

$$= \lambda^2 - 12\lambda + 31$$

$$\lambda^2 - 12\lambda + 31 = 0$$

$$\Rightarrow \lambda = 3.76, 8.236$$

Both eigen values are positive, hence it is a positive definite matrix hence $\bar{x}^*$ is the required minima

AAM

$$\bar{x}^* = \begin{bmatrix} -0.1935 \\ -0.548 \end{bmatrix}$$

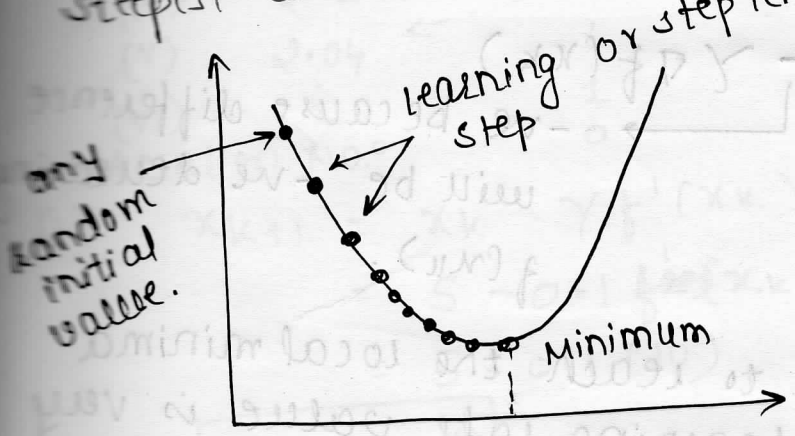
$$\therefore f(\bar{x}^*) = -0.1935 + 2(-0.548) + 4(-0.1935)^2 - (-0.1935 \times -0.548) + 2(-0.548)^2$$

$$f(\bar{x}^*) = 7$$

Ans: function $f(x,y) = x^3 + y^3 + 3xy$.

In practice, computing & storing full Hessian matrix takes large memory which is infeasible for high-dimensional functions. For such situations we use Gradient Descent algorithm.

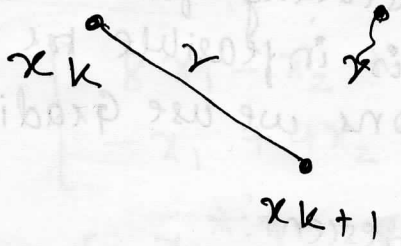
Steepest descent or Gradient descent:



suppose you are at the top of a mountain & want to reach the base camp which is all the way down at the lowest point of the mountain. Due to bad visibility because of weather you cannot see the path at all. How would you reach the base camp? You can use your feet to know where the land tends to descend. This will give an idea ~~about~~ about direction, slope & the length of your step. If you follow the descending path where you are getting -ve slope (gradient) until you encounter a plain area or an ascending path where the gradient becomes +ve, it is very likely you would reach the base camp.

AAM

This algorithm is based on the above mentioned concept. It is basically based on the fact that at any given point x_k, the function $f(x)$ decreases fast in the direction of negative gradient & increases in the opposite direction.



If one goes from x_k to x_{k+1} the step length is γ.
& $\nabla f(x_k)$ is the rate of change of function,

Thus the new point

$$x_{k+1} = x_k - \gamma \nabla f(x_k)$$

-ve because difference will be -ve decreasing $f(x_k)$.

For gradient descent to reach the local minima the selection of this learning rate value is very important. If you select big value as your learning rate, then it may not reach the local minimum because it bounces back & forth between the convex function of gradient descent. If you select very low value of your learning rate then it will take more iteration & hence more time to find the minima.

AAM

Ex:- For given univariate function $f(x) = x^2$, apply gradient descent algorithm.

Solun:- $f(x) = x^2$ let $\gamma = 0.1$ & initial point $x_1 = 5$.

$f'(x) = 2x$

By gradient descent algorithm

$$x_{k+1} = x_k - \gamma f'(x_k)$$

Iteration No:- $x_1 = 5$ $\gamma = 0.1$ $f'(x_1) = 10$.

1 :

$$x_2 = x_1 - 0.1 f'(x_k)$$
$$= 5 - 0.1(10)$$

$\boxed{x_2 = 4}$ $\boxed{f(x_2) = 16}$ $f'(x_2) = 8$

2: $x_3 = x_2 - 0.1 f'(x_2)$

$= 4 - 0.1(8)$

$\boxed{x_3 = 3.2}$ $\boxed{f(x_2) = 10.24}$ $f'(x_3) = 6.4$

3: $x_4 = x_3 - 0.1 f'(x_3)$

$= 3.2 - 0.1(6.4)$

$\boxed{x_4 = 2.56}$ $\boxed{f(x_2) = 6.55}$ $f'(x_4) = 5.1$

4: $x_5 = x_4 - 0.1 f'(x_4)$

$= 2.56 - 0.1(5.12)$

$\boxed{x_5 = 2.04}$ $\boxed{f(x_2) = 4.19}$ $f'(x_5) = 4.08$

5: $x_6 = x_5 - 0.1 f'(x_5)$

$= 2.04 - 0.1(4.08)$

$\boxed{x_6 = 1.632}$ $\boxed{f(x_2) = 2.66}$ $f'(x_6) = 3.264$

Thus we can see that $f(x)$ is reducing, the iteration continues & we get at $x^* = 0$ $f(x^*) = 0$. Thus the algorithm converges to optimum value 0.

Ex2:- For given multivariate function with $x = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ & $f(x) = 4x_1^2 + 3x_1x_2 + 2.5x_2^2 - 5.5x_1 - 4x_2$. Apply gradient descent algorithm. ($\gamma = 0.135$)

Solun:- Since it is multivariate, the variable x is a vector $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$.

$$f(x) = 4x_1^2 + 3x_1x_2 + 2.5x_2^2 - 5.5x_1 - 4x_2$$

&

$$f'(x) = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \end{bmatrix} \qquad f'(x) = \begin{bmatrix} 8x_1 + 3x_2 - 5.5 \\ 3x_1 + 5x_2 - 4 \end{bmatrix}$$

let $x_1 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ $\gamma = 0.135$

By gradient descent $x_{k+1} = x_k - \gamma f'(x_k)$

$x_1 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$

$f'(x_1) = \begin{bmatrix} 8\times 2 + 3\times 2 - 5.5 \\ 3\times 2 + 5\times 2 - 4 \end{bmatrix}$

$f'(x_1) = \begin{bmatrix} 16.5 \\ 12 \end{bmatrix}$

Iteration NO

1:

$$x_2 = x_1 - 0.135 \begin{bmatrix} 16.5 \\ 12 \end{bmatrix}$$

$$= \begin{bmatrix} 2 \\ 2 \end{bmatrix} - 0.135 \begin{bmatrix} 16.5 \\ 12 \end{bmatrix}$$

$$\boxed{x_2 = \begin{bmatrix} -0.2275 \\ 0.38 \end{bmatrix}}$$

$$f'(x_2) = \begin{bmatrix} 8\times(-0.2275) + 3(0.38) - 5.5 \\ 3\times(-0.2275) + 5(0.38) - 4 \end{bmatrix} = \begin{bmatrix} 6.18 \\ -2.70825 \end{bmatrix}$$

$$f(x_2) = 4(-0.2275)^2 + 3(-0.2275)(0.38) + 2.5(0.38)^2 - 5.5(-0.2275) - 4(0.38)$$

$$\boxed{f(x_2) = 0.0399}$$

AAM

2. $x_3 = x_2 - 0.135 \begin{bmatrix} 6.18 \\ -2.16825 \end{bmatrix}$

$= \begin{bmatrix} -0.2275 \\ 0.38 \end{bmatrix} - 0.135 \begin{bmatrix} -6.18 \\ -2.16825 \end{bmatrix}$

$$\boxed{x_3 = \begin{bmatrix} 0.6068 \\ 0.7556 \end{bmatrix}}$$

$$f'(x_2) = \begin{bmatrix} 8 \times 0.6068 + 3 \times 0.7556 - 5.5 \\ 3 \times 0.6068 + 5 \times 0.7556 - 4 \end{bmatrix}$$

$$f'(x_2) = \begin{bmatrix} 1.6212 \\ 1.5984 \end{bmatrix}$$

$$f(x_2) = 4(0.6068)^2 + 3(0.6068)(0.7556) + 2.5(1.5984 - 0.7556)^2 - 5.5(0.6068) - 4(0.7556)$$

$$\boxed{f(x_2) = -2.0841}$$

3. $x_4 = x_3 - \gamma f'(x_3)$

$= \begin{bmatrix} 0.6068 \\ 0.7556 \end{bmatrix} - 0.135 \begin{bmatrix} 0.6068 \\ 0.7556 \end{bmatrix}$

$$\boxed{x_4 = \begin{bmatrix} 0.3879 \\ 0.5398 \end{bmatrix}}$$

$$\boxed{f(x_4) = -2.3342}$$

And the iteration continues, at every iteration the value of the objective function keeps coming down. continuing further we get

$$x_5 = \begin{bmatrix} 0.4928 \\ 0.5583 \end{bmatrix} \text{ & } f(x_5) = -2.3675$$

AAM

After few more iterations we get optimal solution

$$(x^*) = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \quad f(x^*) = -2.3750$$

Thus algorithm converges. convergence can be decided by the uses based on

(i) vector difference between x_k & x_{k+1}

(ii) difference between $f^*(x_k)$ & $f^*(x_{k+1})$ or

(iii) $\nabla f(x^{k+1})$ & $\nabla f(x^k)$. difference. Typically when optimal solution is obtained these differences become 0.

H.W: Minimize $f(x,y) = x^2 + y^2$; $\gamma = 0.1$ $x = 1$ & $y = 1$

AAM