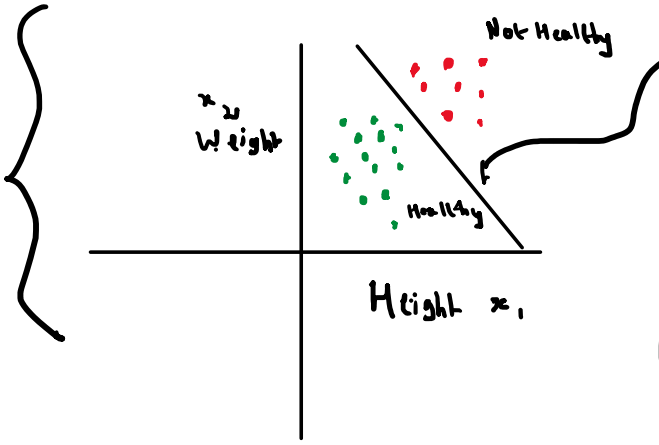
Gradient Descent Recap

03 September 2025 09:12

What will minimizing it do?



$W^T x + W_0 = 0$

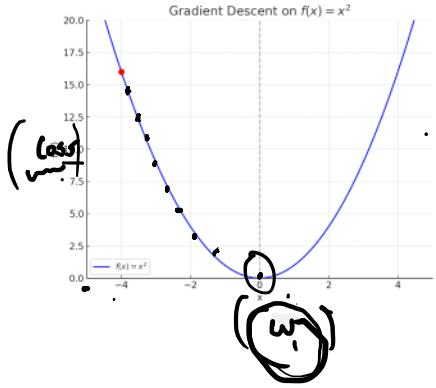
⇓

[optimal weights]

⇓

[align the boundary]

How do we minimize loss fn? ⇒ Gradient Descent



for simplicity, taking $L = w_1^2$

$$w^{t+1} = w^t - \lambda \frac{dL}{dw}$$

λ : learning rate

initialise a random weight

Gradient Descent Run

09 September 2025 19:22

w_1 w_2 w_0

$0.1x_1 - 0.05x_2 + 8.585$ => i) Initialize weights

Student	Blood Sugar (x1 mg/dL)	BMI	y	Predicted ŷ
1	120	28	1	1
2	130	30	1	1
3	115	25	1	-1
4	85	20	1	-1
5	75	18	1	1
6	90	22	-1	1
7	95	24	-1	1
8	100	25	-1	-1
9	80	20	-1	1
10	85	21	-1	-1

$\left\{ = -\sum \left(\frac{w^T x + w_0}{\|w\|} \right) y_i \right\}$

$\left\{ \text{Loss} = -\sum \left(\frac{w^T x + w_0}{\|w\|} \right) y_i \right\}$

Student	Blood Sugar (x1 mg/dL)	BMI	y	Predicted ŷ	Loss
1	120	28	1	1	0
2	130	30	1	1	0
3	115	25	1	-1	99.0
4	85	20	1	-1	74.2
5	75	18	1	1	0
6	90	22	-1	1	79.1
7	95	24	-1	1	84.1
8	100	25	-1	-1	0
9	80	20	-1	1	70.7
10	85	21	-1	-1	0

$\approx$ [individual losses]

[How many computations? Per update

10 or n where n is no. of datapoints

$0.1x_1 - 0.05x_2 + 8.585$ => $w_1x_1 + w_2x_2 + w_0$

[Total Loss = 407.1] → Adding up losses of how Many points - 10 points

$w_1 = 0.1$

$w_2 = -0.05$

$w_0 = 8.585$

{What happens next in order to Minimize loss?}

$$(w_1, w_2, w_0)^{new} = (w_1, w_2, w_0)^{old} - \eta\{\nabla l\}$$

After w_1, w_2, w_0 are updated

$$-0.35x_1 - 0.162x_2 + 8.580$$

$$w_1 = -0.35$$
$$w_2 = -0.162$$
$$w_0 = 8.58$$

Student	x1	x2	y	f_new	ŷ_new	loss
1	120	28	1	-0.35120 -0.16228 +8.58 = -37.956	-1	37.956
2	130	30	1	-45.06	-1	45.06
3	115	25	1	-35.955	-1	35.955
4	85	20	1	-26.64	-1	26.64
5	75	18	1	-23.346	-1	23.346
6	90	22	-1	-0.3590 -0.16222 +8.58 = -26.484	-1	0
7	95	24	-1	-28.338	-1	0
8	100	25	-1	-30.03	-1	0
9	80	20	-1	-24.86	-1	0
10	85	21	-1	-26.062	-1	0

$$\text{loss} = 37.956 + 45.06 + 35.955 + 26.64 + 23.346 = 168.957$$

How many times will I repeat this ?

How many times will I repeat this !

{ Until I feel I have reached a Minima }

{ [Vanilla / Batch Gradient Descent] }

Stochastic Gradient Descent

09 September 2025 21:26

Randomly initialize weights (1000 times

Student	Blood Sugar (x1 mg/dL)	BMI	y	Predicted ŷ	Loss
1	120	28	1	1	0
2	130	30	1	1	0
3	115	25	1	-1	99.0
4	85	20	1	-1	74.2
5	75	18	1	1	0
6	90	22	-1	1	79.1
7	95	24	-1	1	84.1
8	100	25	-1	-1	0
9	80	20	-1	1	70.7
10	85	21	-1	-1	0

{Dont take overall loss}

1 computation

{ $w_1 = 0.01$, $w_2 = 0.5$, $w_0 = -9$ }

$w_{new} = w_{old} + \eta \nabla l \Rightarrow$ $\left(-\left[\frac{w^T x + w_0}{\|w\|} * y_i\right]\right)$

{loss of only one point}

{1 computation per weight update}

Stochastic Gradient Descent

Stochastic → Randomness/simulation

[Will we still get similar result?]

[Yes we will get a similar result]

{G.D ⇒ Access to all points} My entire dataset

1. Batch Gradient Descent

- Computes the gradient of the loss function across the entire dataset:

$$\nabla_\theta L(\theta) = \sum_{i=1}^{N} \nabla_\theta \ell(x_i, y_i; \theta)$$

- This is precise because you're using all data points, but for **large datasets**, computing this sum at every step is expensive.
- Each update happens **after seeing the whole dataset**, so the model updates less frequently.

2. Stochastic Gradient Descent

- Uses only **one data point** (or a small batch) to estimate the gradient:

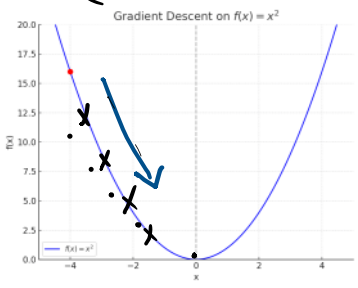
$$\nabla_\theta \ell(x_i, y_i; \theta)$$

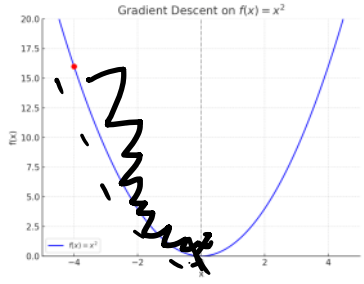
- The gradient is **noisy** because it's based on one sample rather than the whole dataset.
- But it's **much faster per update** because you avoid summing over N points.

(single points)

Batch G.D

Stochastic



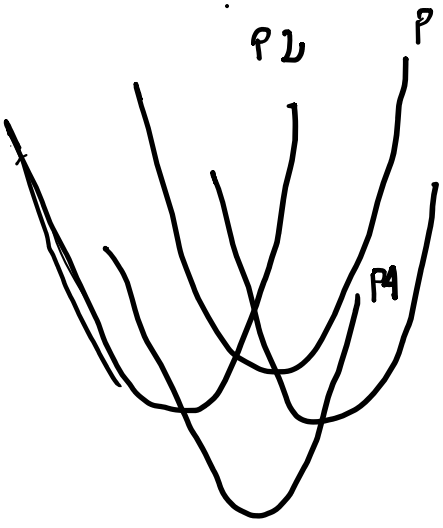


individual points have individual loss shapes

If I need to update weights 100 times for regular G.D,

Variety into gradient descent

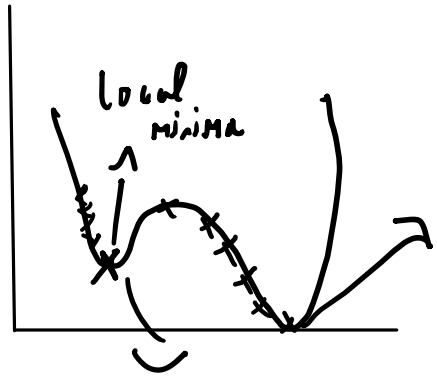
For stochastic, I will need to do



=> yes { Variety is precious }

Learn better

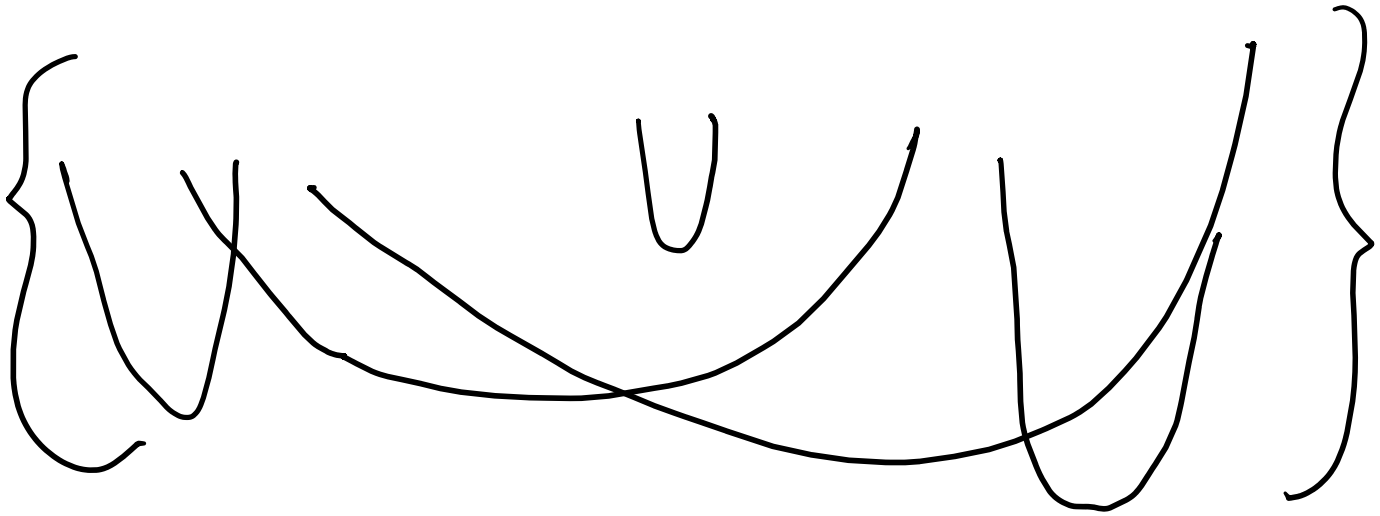
It {might} help us solve the [non-convex problem]



{is escaped}

{Disadvantage ?} Double edged sword

When each datapoint is too different from each other



{Variety can backfire}

Chance of missing global Minima

If 100 points,

(Vanilla / Batch GD) ↳ n = 100 points per weight update

(Stochastic G.D) ↳ 1 point per weight update

Subset of the points

10 points at a time } :) randomly

$$-\left\{\sum_{i=1}^{10}\left(\frac{W^T x + W_0}{\|w\|}\right) y_i\right.$$

{ Mini- Batch Gradient Descent }

{Best of both worlds}

Interview :-

i) Loss ✓
ii) Speed ✓
iii) Noise ✓

Epoch vs iterations

10 September 2025 06:52

Student	Blood Sugar (x1 mg/dL)	BMI	y	Predicted ŷ
1	120	28	1	1
2	130	30	1	1
3	115	25	1	-1
4	85	20	1	-1
5	75	18	1	1
6	90	22	-1	1
7	95	24	-1	1
8	100	25	-1	-1
9	80	20	-1	1
10	85	21	-1	-1

100

(Interview)

(Epoch)

{One round of visiting all points in My dataset}

{100 points ~ Stochastic Gradient Descent} (1 epoch)

How many updates per epoch => (100) ✓

Mini batch Gradient Descent

Batch size = 10 ✓

10 batches

x_1 to x_{10}

x_n to x_{20}

x_{20} to x_{30}

{How many updates per epoch -> 10}

Iterations -> Number of times the

Weight is updated

Interview :-

{ Difference b/w epoch and iteration }

2 epochs

If my batch size is 10

{ 20 updates / Iterations }

Coding Gradient Descent

11 September 2025 00:01

https://colab.research.google.com/drive/14aBoDBQIy6rRAoajnvnf1Ct-JgTO--uj?authuser=1#scrollTo=c157b062-2759-4cc0-aed0-333eb871838e

How can I minimize a function $[3x^2+4]$?

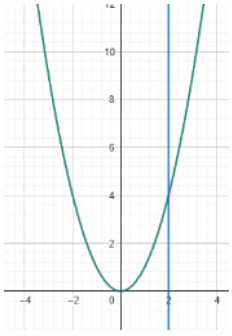
$$\frac{d(3x^2+4)}{dx} = 0$$

$$6x = 0$$

$$x = 0$$

$$f(x) = 3 \times 0 + 4 = 4$$

{ Minimize $f(x)=x^2$ such that $x=2$ }



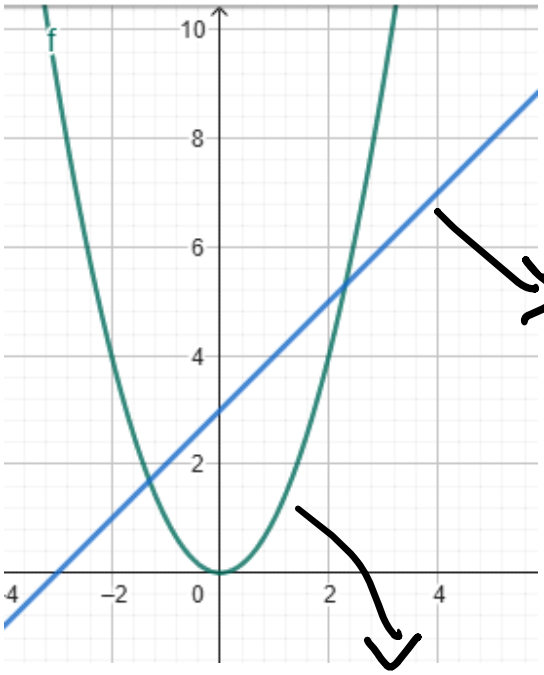
Minimize $f(x) = x^2$ such that

$$y = x + 3$$

$$x^2 = x + 3$$

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

$$\left\{ \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \right\}$$



Objective

Minimize $(x^2 + y^2) \Rightarrow f(x)$

$(x + y - 2 = 0) \Rightarrow g(x)$

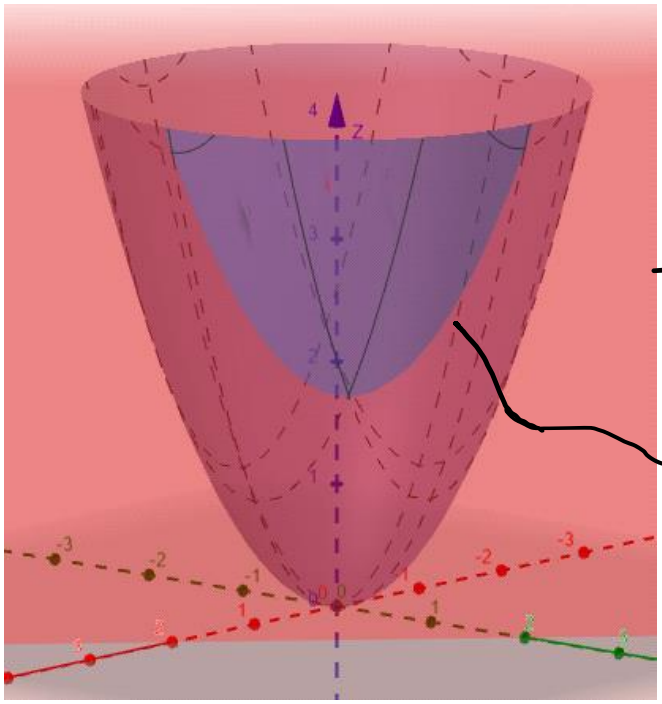
Instead of substituting.

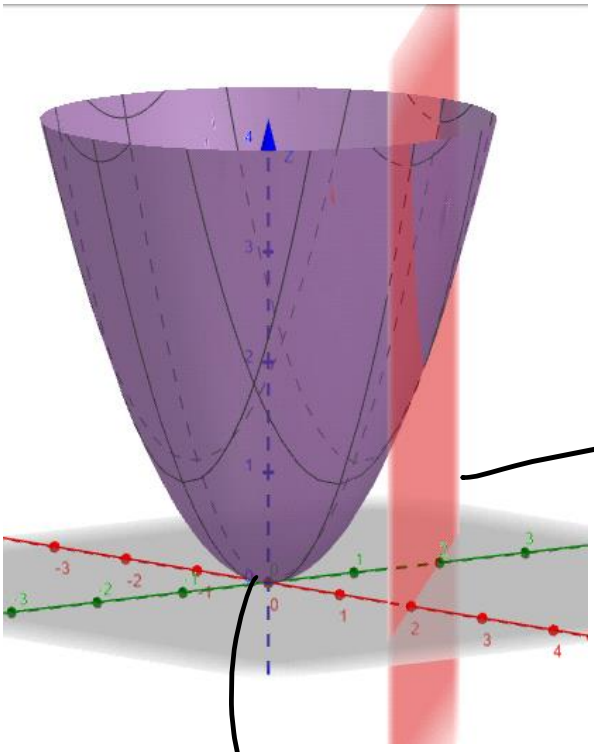
Add the constraint to the objective function.

$\{\lambda = 0\} \Rightarrow$ enforce the constraint $\Leftrightarrow$ how much to enforce?

$\{$ Minimize $\{f(x) + \lambda\, g(x)\}$ $\}$

$\{$ Lagrange Multiplier $\}$


Constraint
Objective


Constraint
objective

x, y, λ

$$x^2 + y^2 - \lambda (x + y - 2)$$

$$1^2 + 1^2 - 2 (1 + 1 - 2)$$

$$(1 + 1 - 2 \times 0 = 2)$$

[Minimum value is $Z = 2$ or $f(x,y) = 2$] where $x, y = (1, 1)$

(Lagrangian multipliers ?

Where ?)