Linear Regression.

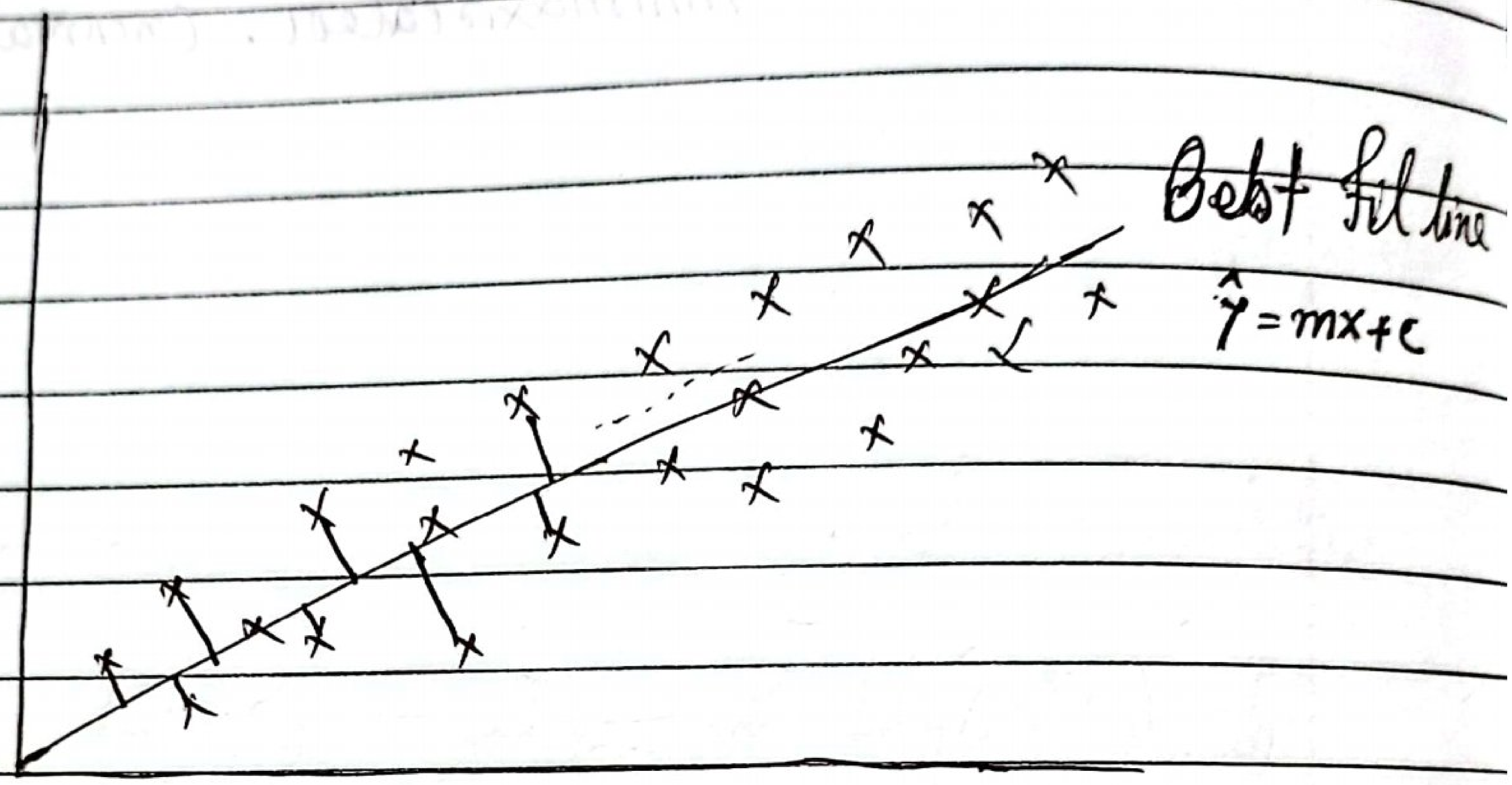
$Y = mx + c$

when $x = 0$

$Y = c$

m = slop [Gradient / Descent.]

c = Intercept



$$\frac{1}{2m}\sum_{i=1}^{m}\left(\hat{y}-y^{i}\right)^2$$

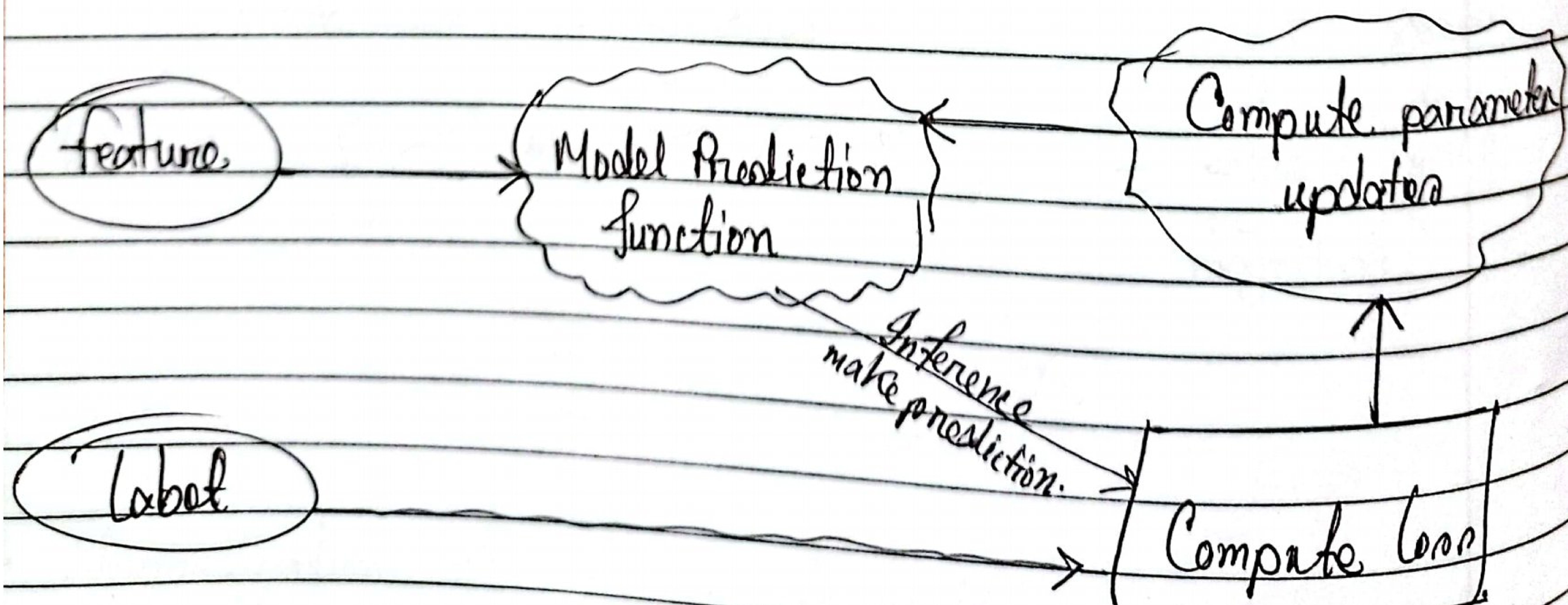
Predicted Point. — Real point.

number of point's.

→ Mean Absolute error

→ Mean square error.

→ Cost function.



Iterative approach.

$\hat{y} = mx + c$

$= 1(1)$

$\hat{y} = 1$

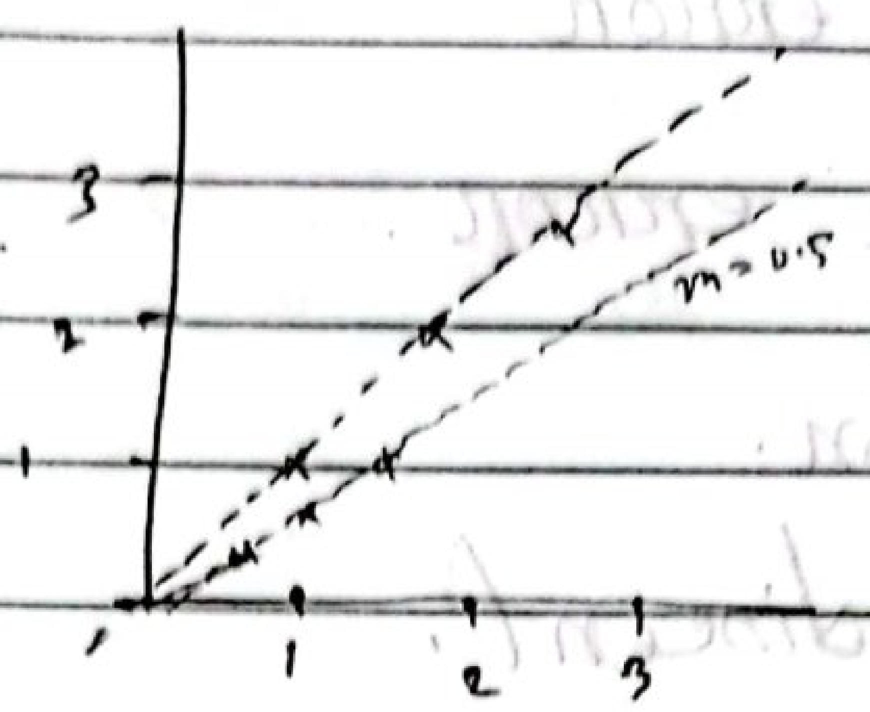
$\hat{y} = 2$

$\hat{y} = 3$

$m = 1$

$x = 1, 2, 3 \ldots$

$c = 0$

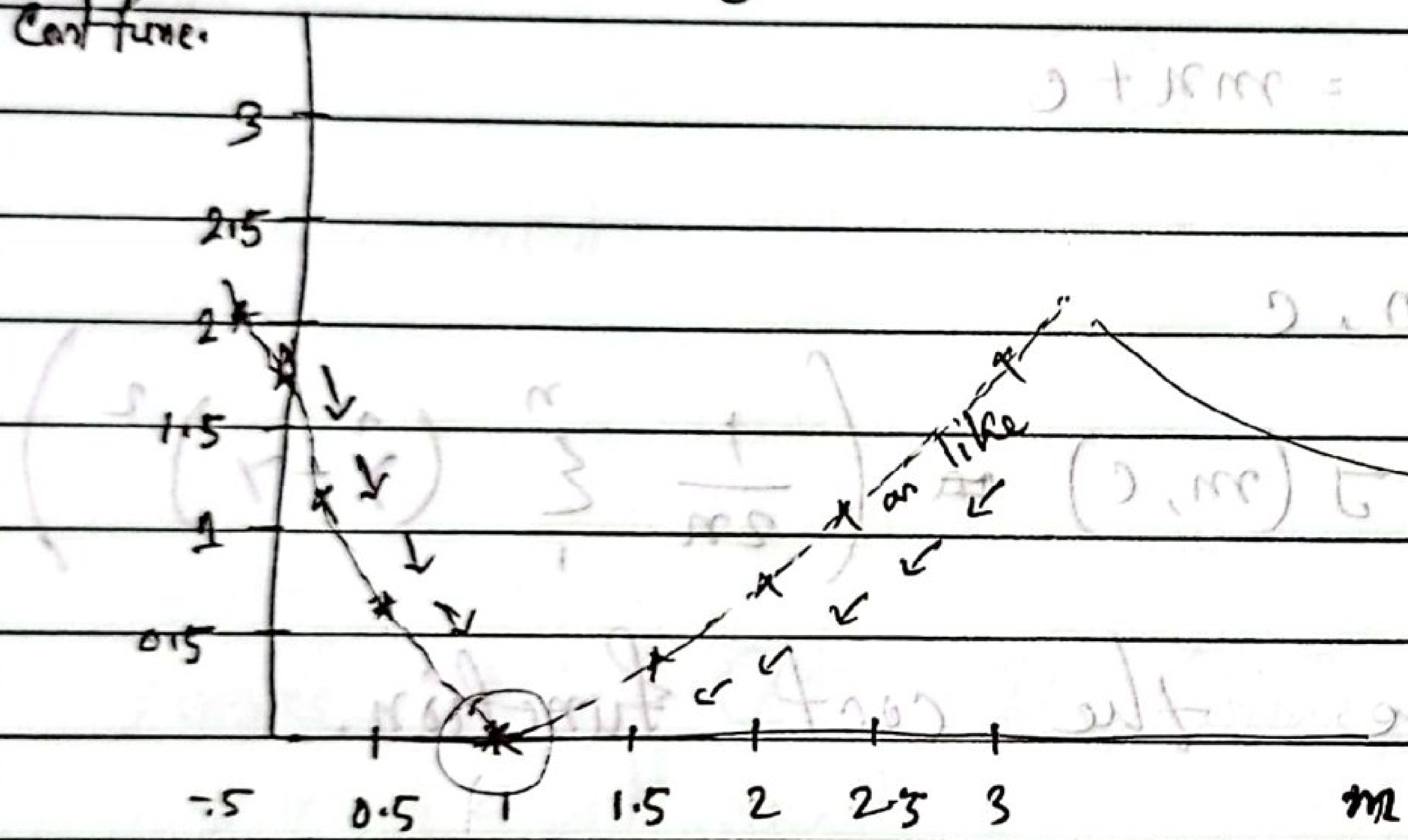


$m = 0, 0.5, 1.0 \ldots$

$c = 0$

$$\text{Cost function} = \frac{1}{2 \times 3} \times \left[(1-1)^2 + (2-2)^2 + (3-3)^2\right]$$

$$= 0$$



Learning Rate

Gradient descent

Step size.

google.developer.com

$m = 1$

Cost $f = 0$

Global minimum.

Converge theorem.

$$m = m - \frac{\partial(m)}{\partial m} \times \alpha$$

Learning rate.

$m = 0.5$

Left $m = m - (-ve) \times$ small value

$= m + ve \times$ small value.

Right $m = m - \overset{(+v)}{\ }$ small value.

$\hat{y} = 0.5 \times 1 = 0.5$

$\hat{y} = 0.5 \times 2 = 1$

$\hat{y} = 0.5 \times 3 = 1.5$

Learning Rate

$\hookrightarrow$ small value.

$\rightarrow$ not too small

$\rightarrow$ not too large

$$\frac{1}{2m}\left((0.5-1)^2 + (1-2)^2 + (1.5-3)^2\right)$$

$$= \frac{1}{6} \times \left(0.25 + 1 + (-1.5)^2\right) = 0.58$$

→ Mean Absolute error → $MSE = \frac{1}{n}\sum_{i=1}^{n}(Y-\hat{Y})^2$

→ Mean Square error. → $C.F = \frac{1}{2n}\sum(\hat{Y}-Y)^2$

→ Cost function.

→ ~~Gradient descent.~~

→ Learning Rate. (α)

→ Hypothesis $\hat{Y} = mx + c$

Parameter m, c

Cost function $J(m,c) = \left(\frac{1}{2n}\sum_{1}^{n}(\hat{Y}-Y)^2\right)$

Goal = minimize the cost function.

Repeat

H-T $h_\theta(x) = \theta_0 + \theta_1 x$

Parameters θ_0, θ_1

Cost function $J(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$

Goal : minimize $J(\theta_0, \theta_1)$

→ Start with θ_0, θ_1 and keep changing them to reduce $J(\theta_0, \theta_1)$ untill we hopefully end up at a minimum.

Gradient descent algorithm.

repeat untill convergence{

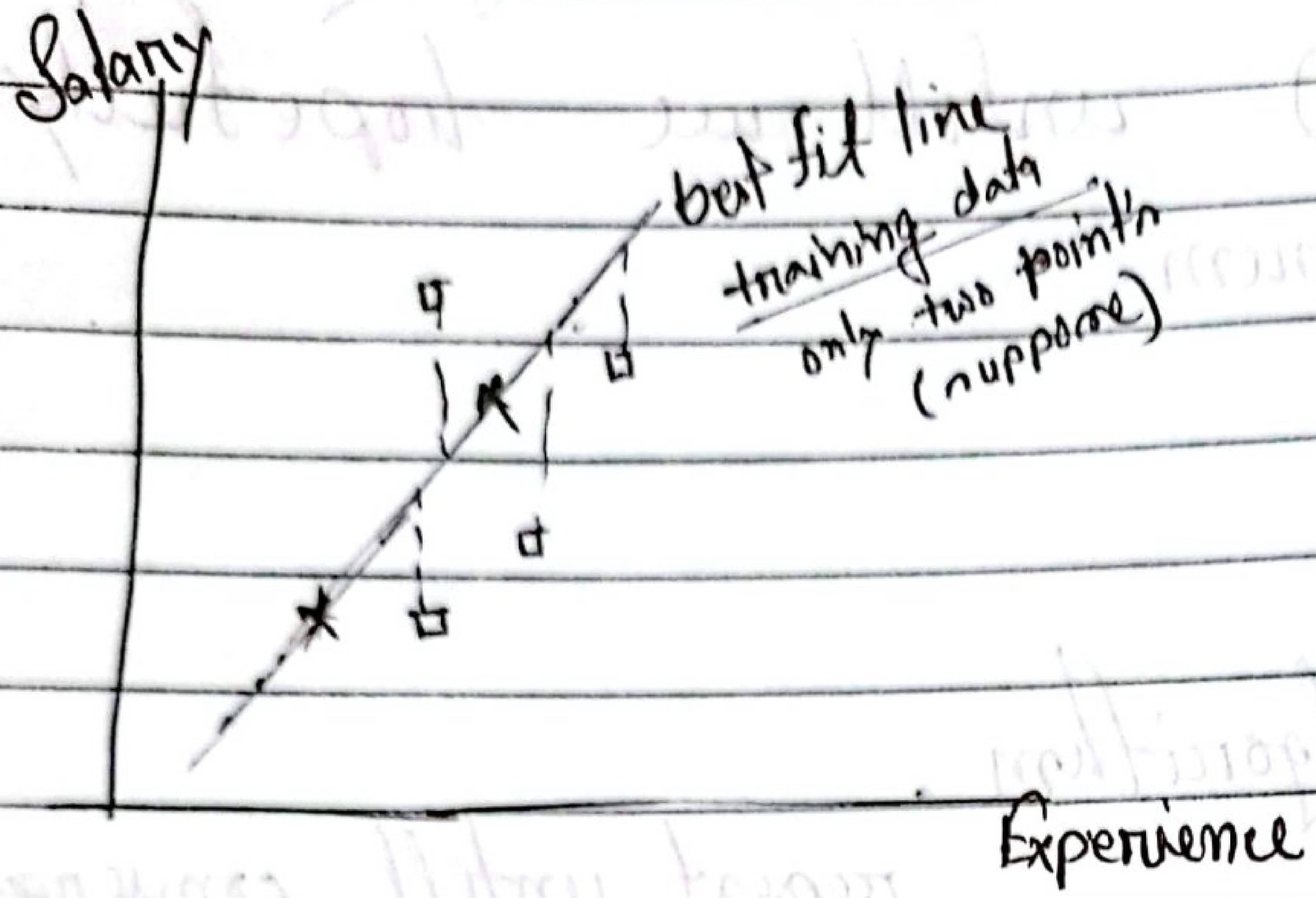
$$\theta_j := \theta_j - \frac{\partial}{\partial \theta_j} \left(J(\theta_0, \theta_1) \right) \times \alpha \}$$

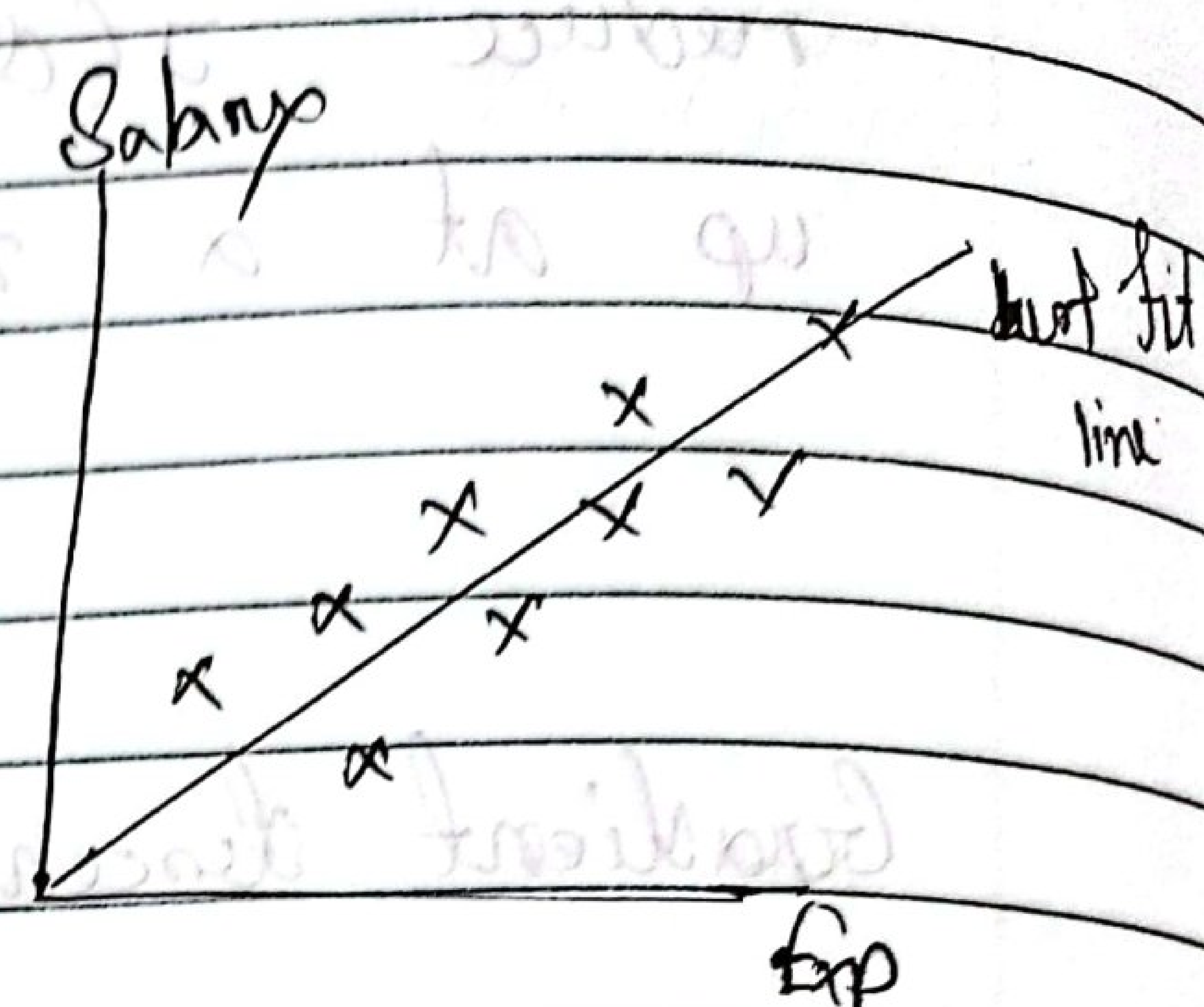
$J=0, \; J=1$

Learning rate

Andrew NG ~~10~~ ~~A~~ Linear Regression term and details summary in "Krish Naik" video.

Ridge and Lasso Regression.





Cost function or
Sum of Residuals $\sum_{i=1}^{n}(y-\hat{y})^2 = 0$

In linear Regression.

$\hat{y} = mx + c$

but for testing data (T) high error.

Overfitting → for training low error
→ " testing high "

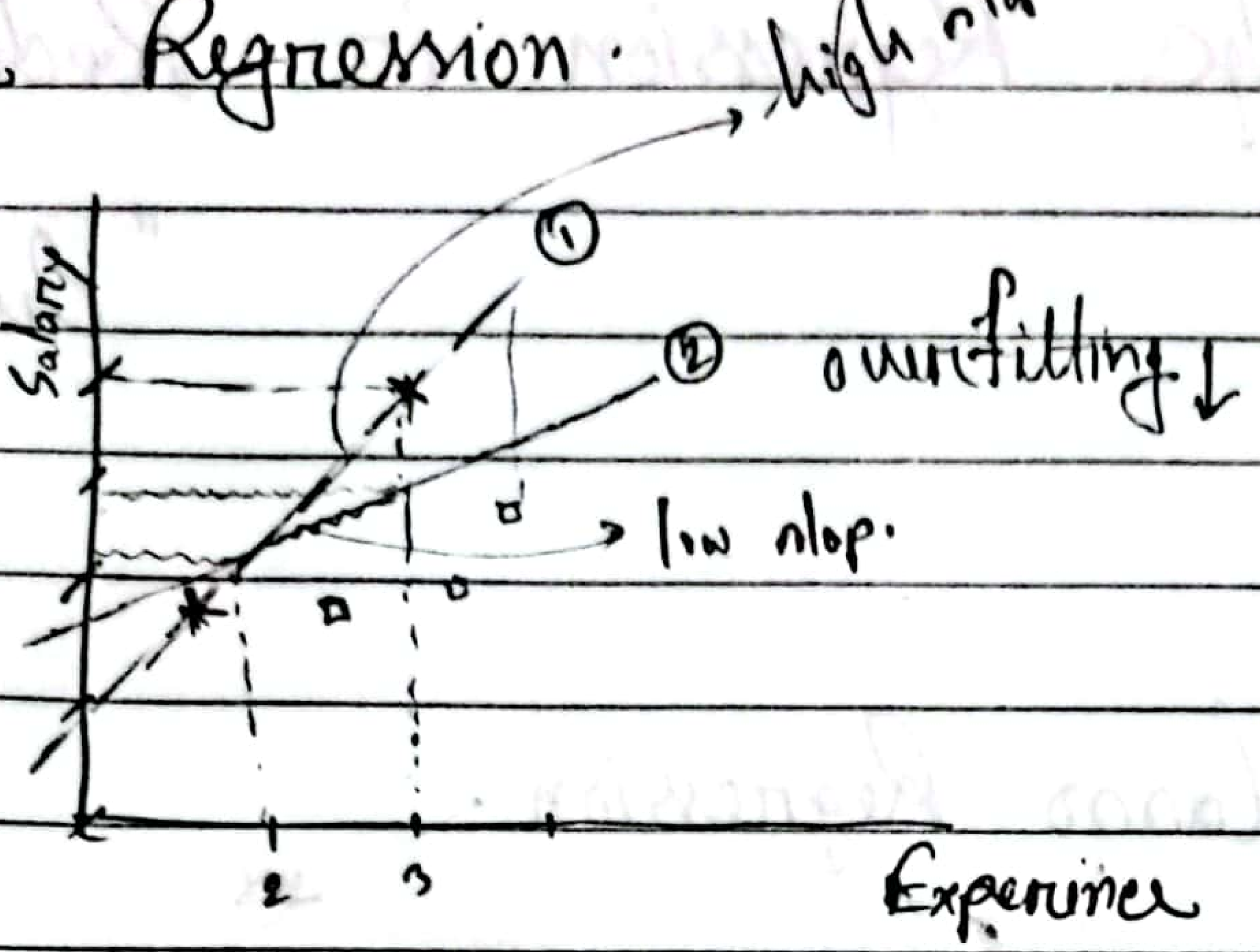
high variance
convert
low variance

→ Underfitting → for training data get high error and for the testing dataset getting high error.

Generalized or Good Model
Should always low bias and low variance.

By Ridge and Lasso Regression

Ridge Regression.



Cost function →

$$\sum_{i=1}^{n} (y-\hat{y})^2 + \lambda(\text{slope})^2$$

Steep slope

খাড়া

① x-এ single unit of change y-এ change অনেক বেশী slope

~~target → slop যত low করা যায়~~

② Small change in x- axis small chang in y- axis .. slope

আরেও ছোট হবে slope.

slop = 1

① Lets, $\lambda = 1$ ধরে নিলাম

slop = ~~1.69~~ = ~~1.69~~ 1.3.

Cost → 0 + 1 × 1.69

= 1.69

② Small value + 1×1^2

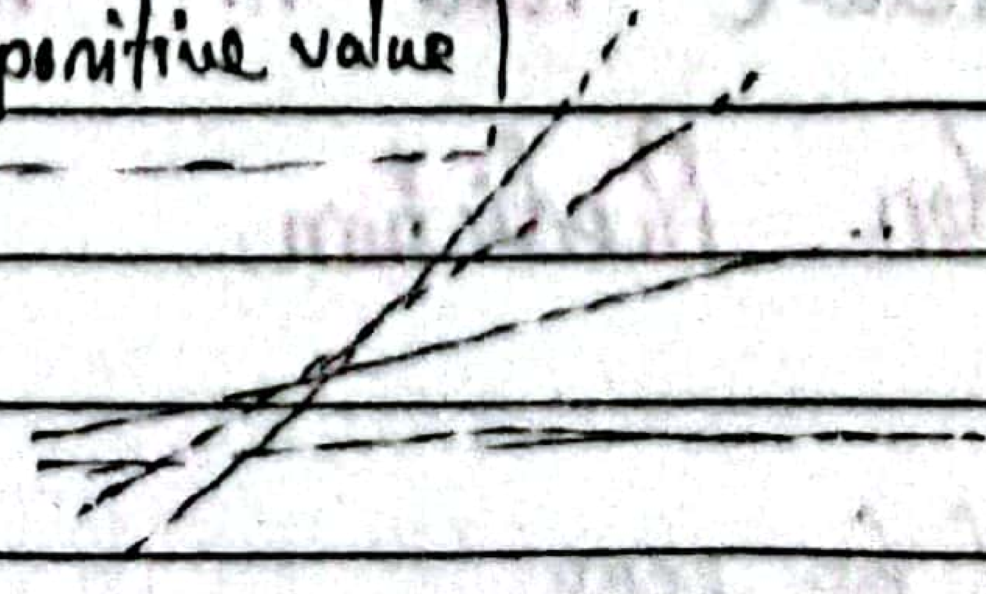
→ 1.39 like.

∴ ~~1.69 > 1.39~~ $\boxed{1.69 > 1.39}$

② num best fit line is better than ①

$\boxed{\to \text{Selected}}$

$\boxed{\lambda > 0 \text{ to any positive value}}$



tends to zero.

not to zero

Main Aim of Ridge Regression is Reducing "Overfitting".

Lasso Regression. → use → Feature Selection

Cost function.

$$\sum_{i=1}^{n} (y-\hat{y})^2 + \lambda \times |slope|$$

magnitude of slop.

⇒ some feature will Remove

$$y = m_1 x_1 + m_2 x_2 + \cancel{m_3 x_3} \dots + c$$

$$\lambda |m_1 + m_2 + m_3 + m_4 \dots|$$

slope value is very very less those feature Remove

Moving towards zero --------> ----> 0

and feature will Remove.

because that is not important for Prediction.

→ once a time it will be zero