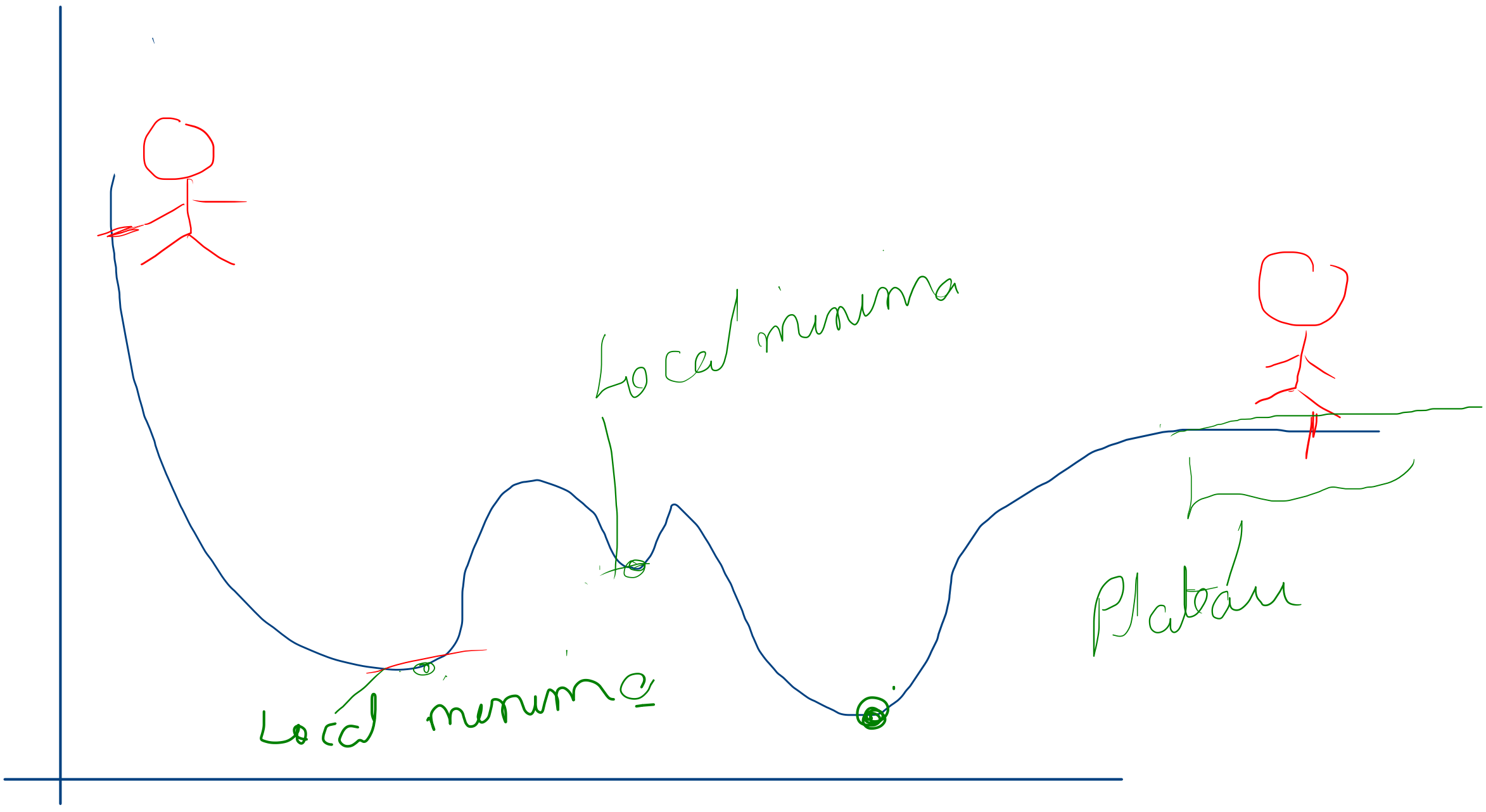
Starting understanding gradient descent.



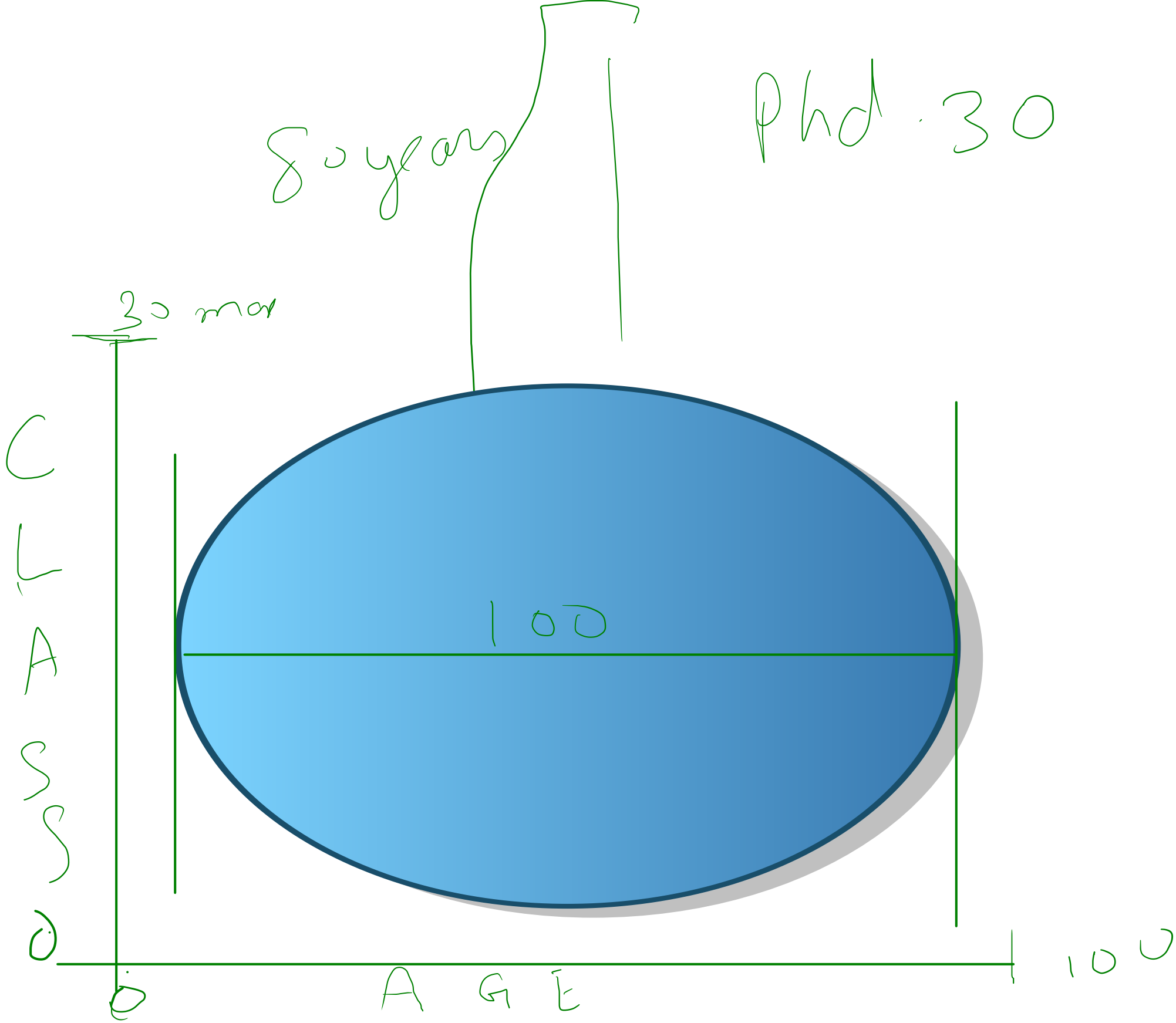
Types of gradient descent :->

1) Batch
2) Stochastic
3) Mini-Batch. Gradient Descent.

But before this we have one more topic to discuss

What would happer if our data is on different scale - or features have different scales?

Student Age	Student class
6	1
16	15 or 16
20	15
55	30


8 years
phd 30
30 mar
CLASS
100
0
AGE
100

2 ways by which we can scale our data.

1) Standard Scaler / Standardization

2) Min-max Scaling / Normalization

Standardization:

$$x_{i\,new} = \frac{x - \mu}{\sigma}$$

where μ = mean

σ = standard deviation.

$$\mu = \frac{1}{n} \sum_{i=1}^{n} x_i$$

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - \mu)^2}$$

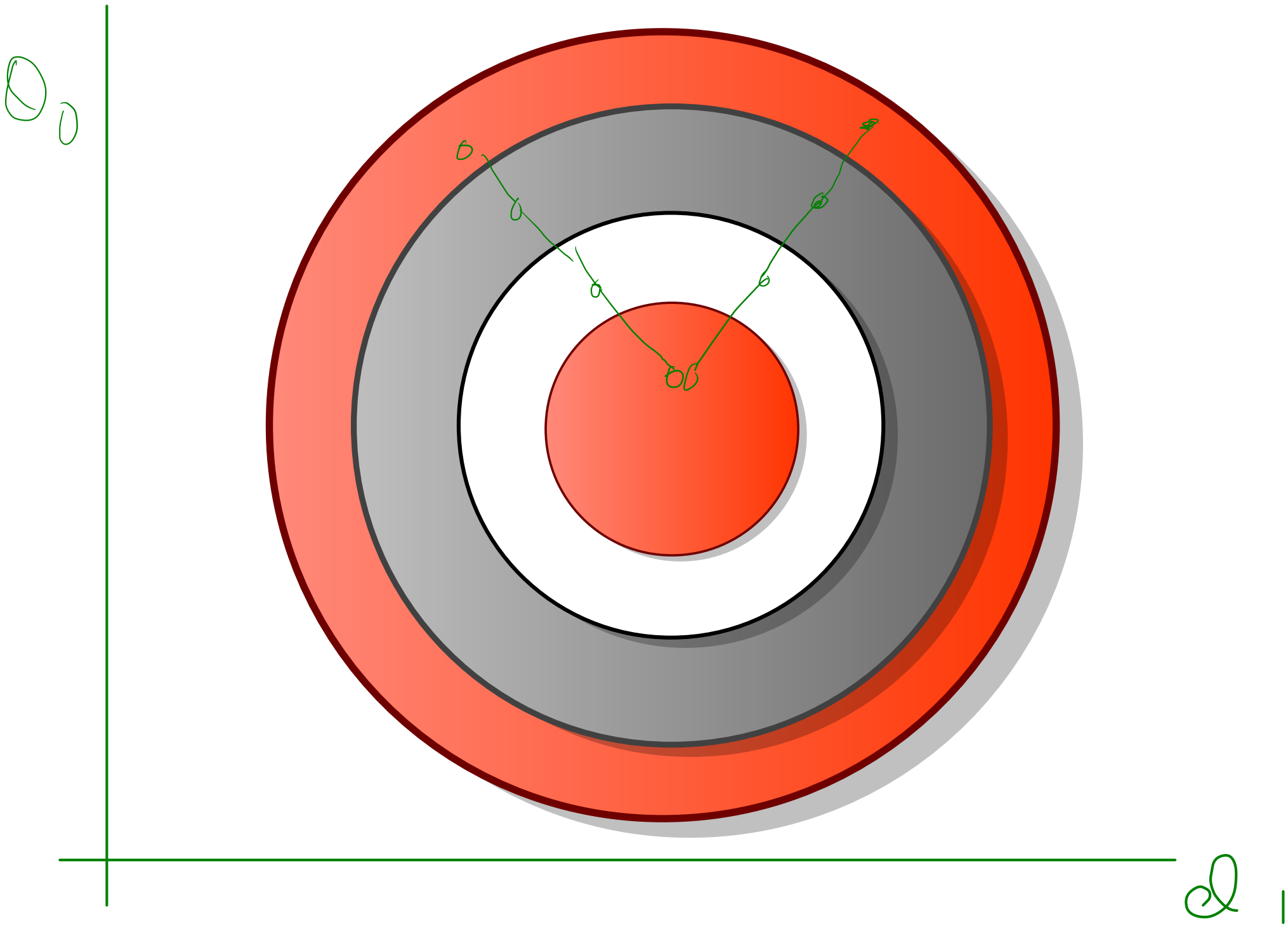
Standardization will scale our data such that mean of data would be zero and standard deviation will be 1 for all the data

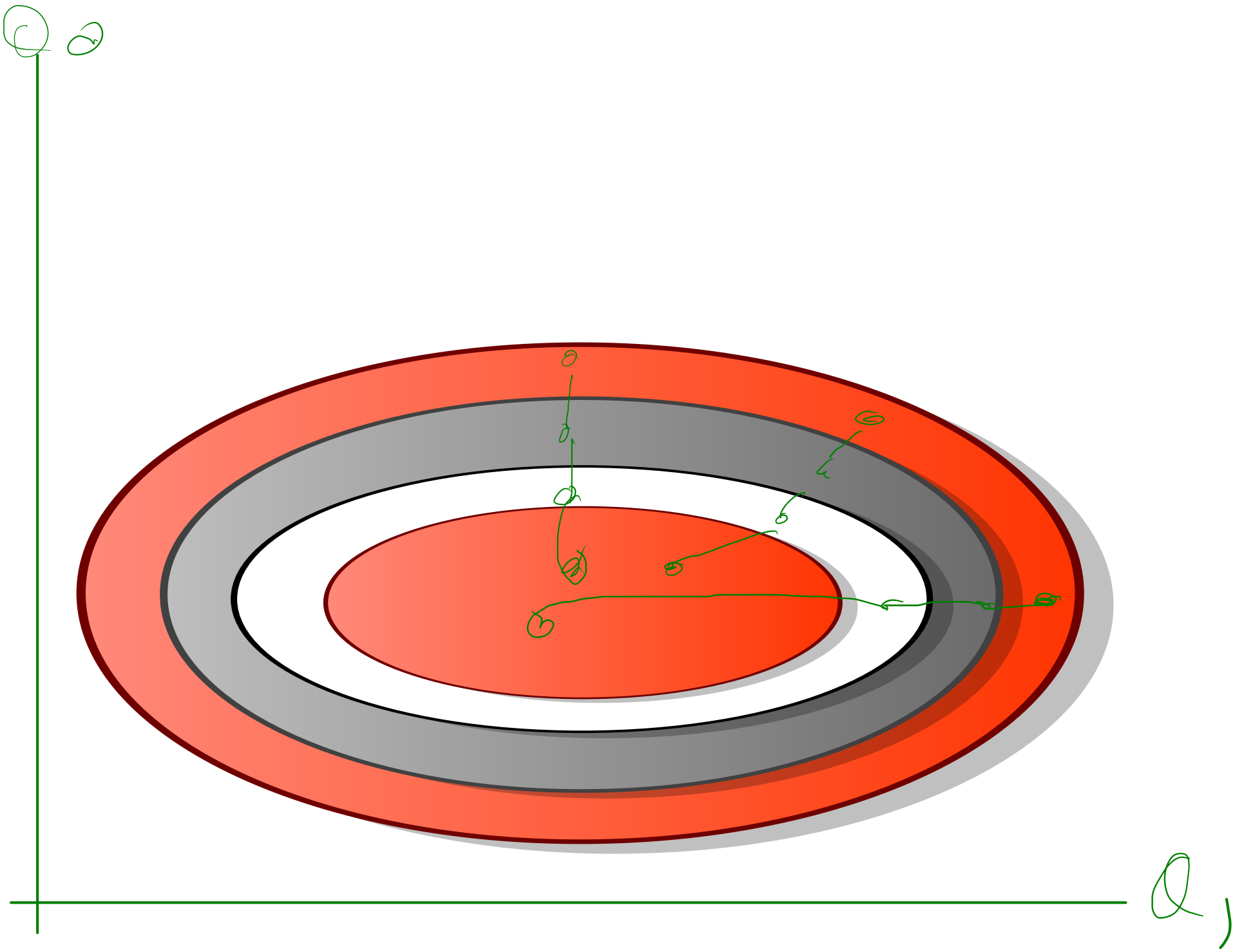
Normalization or Min max scaling

$$x_{i\,new} = \frac{x_i - min}{max - min}$$

The data will range b/w - 0 & 1.

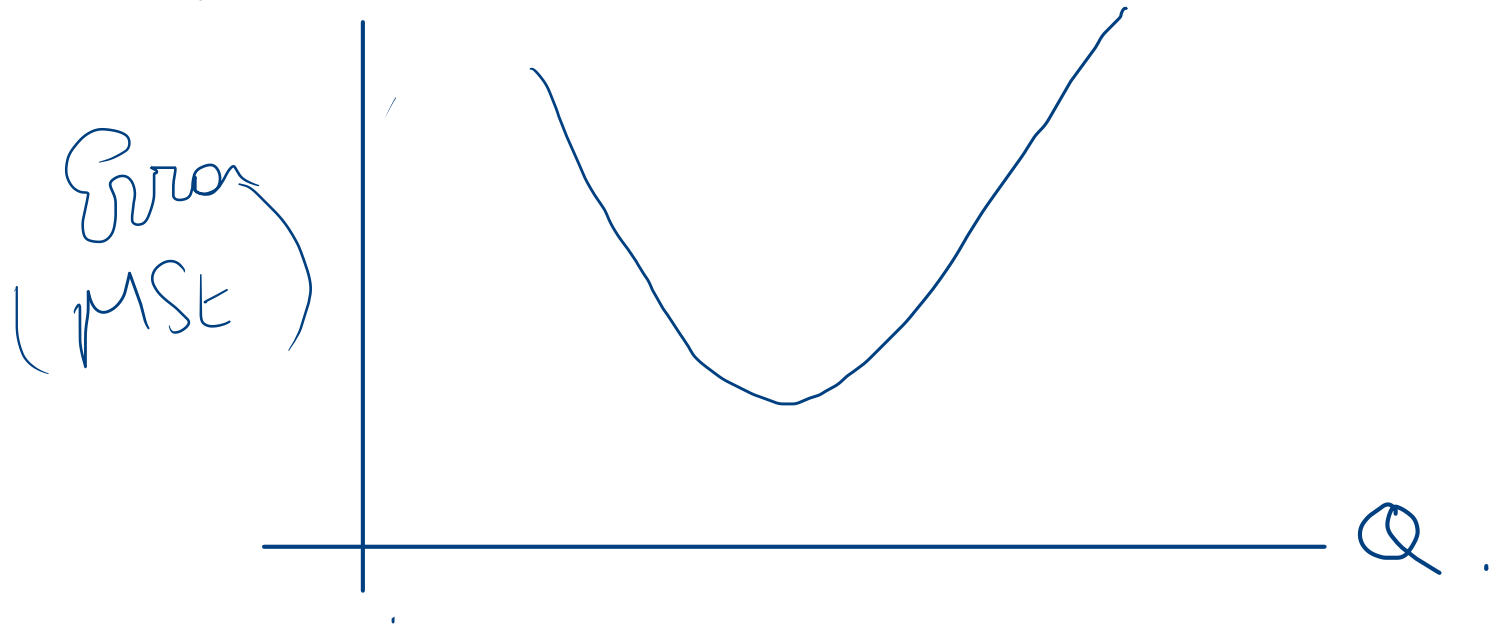
Now we will see the effect of scaling on gradient descent.





What do we mean by training a model?

We are searching for best parameters
- for $\theta_0, \theta_1 \dots \theta_n$ To minimize
cost function (like MSE)



Model becomes complex as the number of features increases.

Types of Gradient descent:

1) Batch gradient descent.

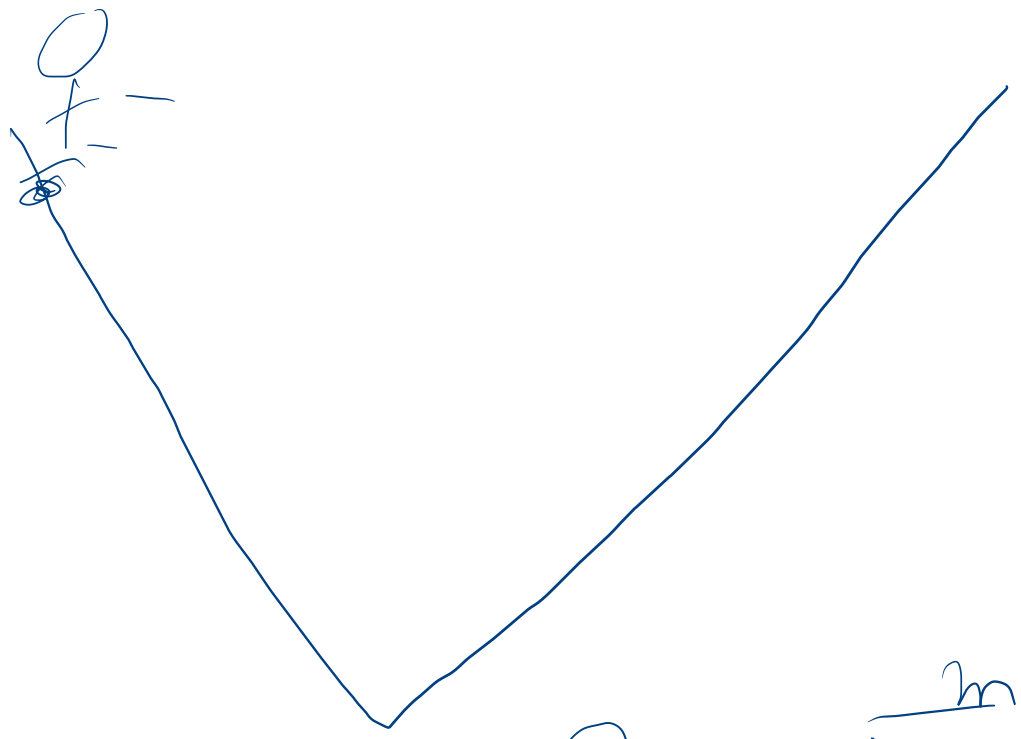
2) Stochastic " "

3) Mini batch " "

Partial Derivative :

First we initialize model parameter $(\theta_1, \theta_2 \dots \theta_n)$ randomly.

For each weight we change one weight w_i & fix all other. Then we check change in Cost function. This is called partial derivative



$$\frac{\partial}{\partial \theta_j}(MSE(\theta)) = \frac{2}{m}\sum_{i=1}^{m}(\theta^T \cdot x^{(i)} - y^{(i)})\, x_j^{(i)}$$

gradient vector $\nabla_\theta MSE(\theta)$

$$= \begin{pmatrix} \frac{\partial}{\partial \theta_0} MSE(\theta) \\ \frac{\partial}{\partial \theta_1} MSE(\theta_1) \cdots \\ \frac{\partial}{\partial \theta_n} MSE(\theta_n) \end{pmatrix}$$

$$= \frac{2}{m} X^T \cdot (X \cdot \theta - y)$$

Since Batch gradient descent is using complete dataset ~~by~~ every step. It is really slow on training. But still faster than linear regression.

Gradient descent

θ (new step) = θ (step) $- \eta \nabla_\theta MSE(\theta)$

learning rate.