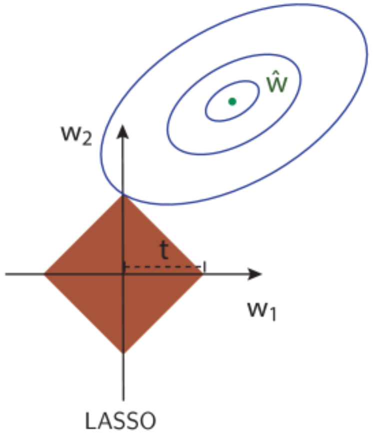
Siddhardhan

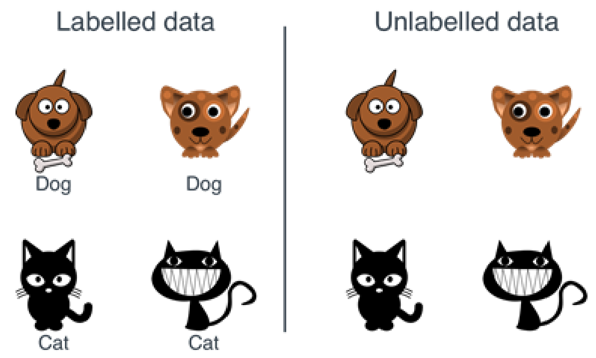
Building Lasso Regression from Scratch in Python

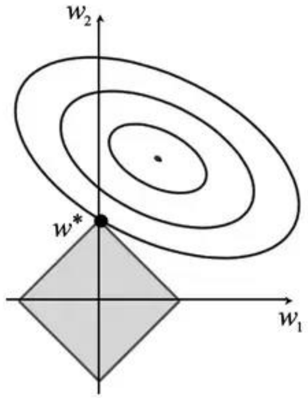


Lasso Regression

About Lasso Regression:

1. Supervised Learning Model
2. Regression model
3. **L**east **A**bsolute **S**hrinkage and **S**election **O**perator
4. Implements Regularization (L1) to avoid Overfitting





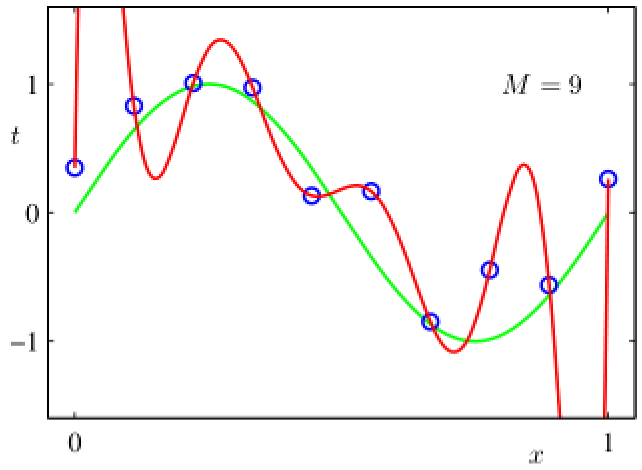
L1

Regularization

Regularization is used to reduce the overfitting of the model by adding a penalty term (λ) to the model. Lasso Regression uses L1 regularization technique.

The “penalty” term reduces the value of the coefficients or eliminate few coefficients, so that the model has fewer coefficients. As a result, overfitting can be avoided.

3rd order Polynomial equation : $y = ax^3 + bx^2 + cx + d$



This Process is called as Shrinkage.

LASSO --> Least **A**bsolute **S**hrinkage and **S**election **O**perator

Lasso Regression

Cost Function for Lasso Regression :

$$J = \frac{1}{m}\left[\sum_{i=1}^{m} \left(y^{(i)} - \hat{y}^{(i)} \right)^2 + \lambda \sum_{j=1}^{n} w_j \right]$$

m --> Total number of Data Points

n --> Total number of input features

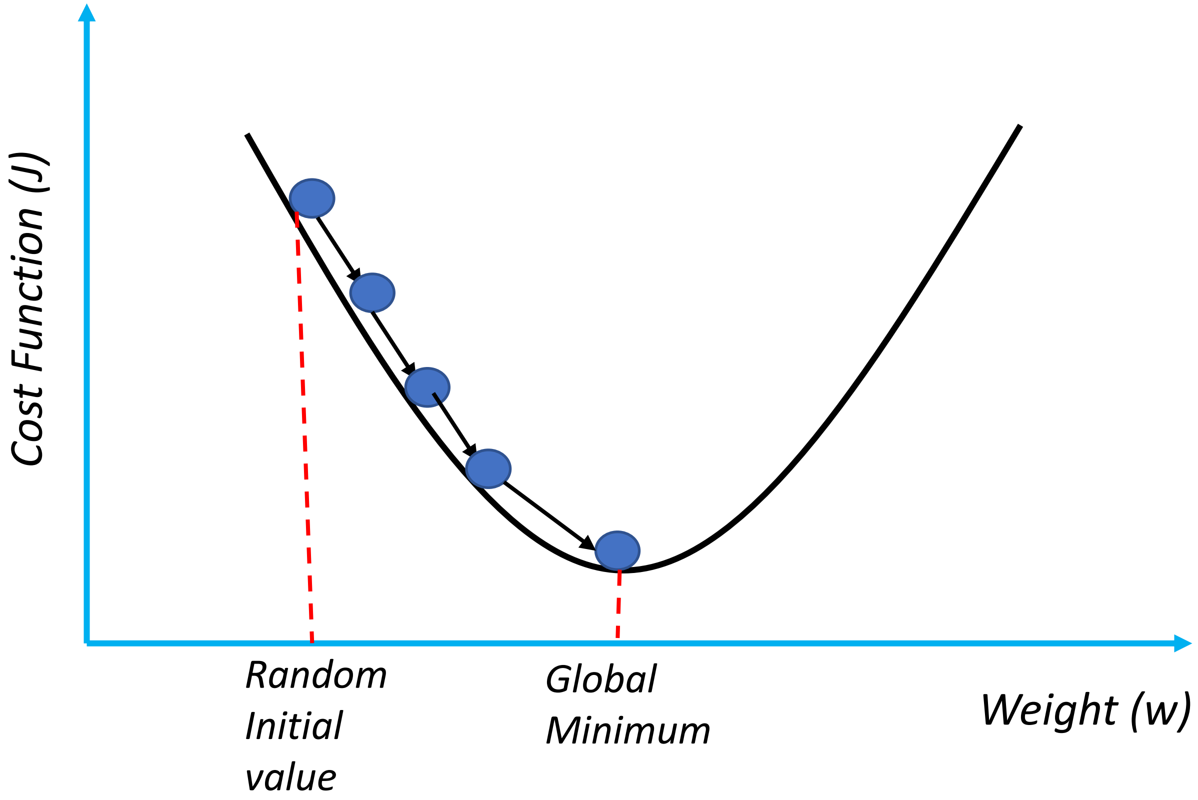
$y^{(i)}$ --> True Value

$\hat{y}^{(i)}$ --> Predicted Value

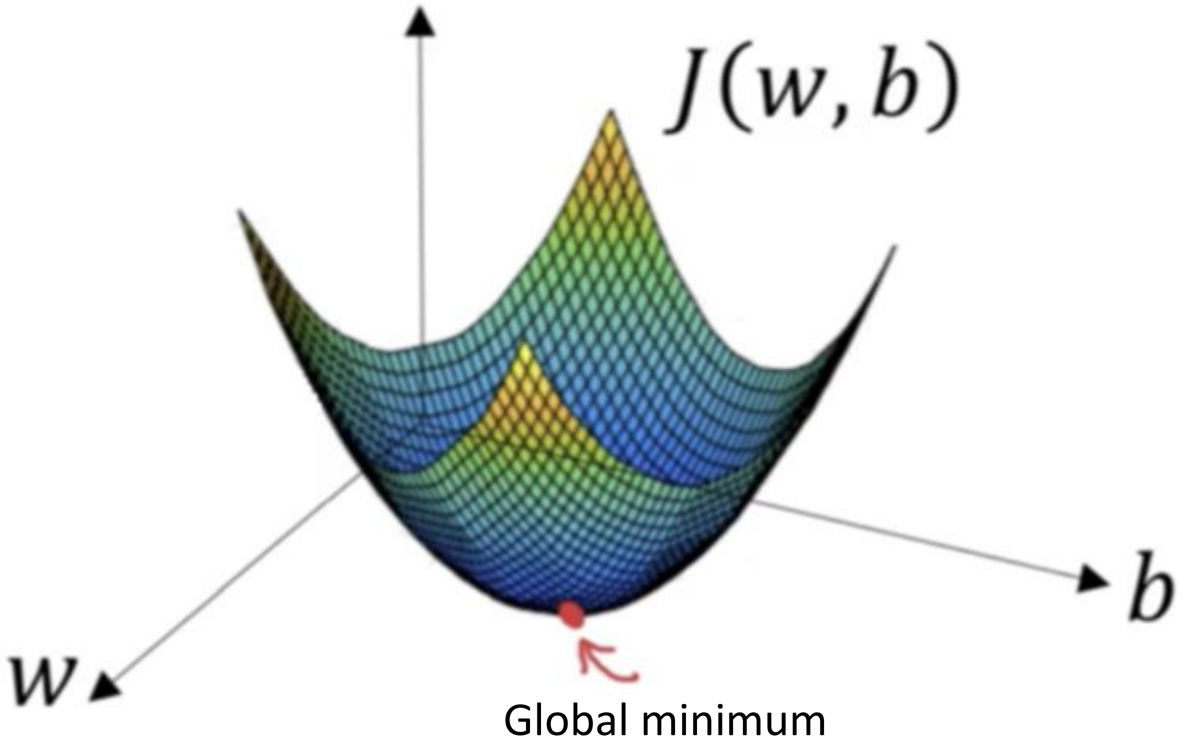
λ --> Penalty Term

w --> Parameter of the model

Gradient Descent



Gradient Descent in 3 Dimension



Gradient Descent

Gradient Descent is an optimization algorithm used for minimizing the cost function in various machine learning algorithms. It is used for updating the parameters of the learning model.

$$w_2 = w_1 - L * \frac{dJ}{dw}$$

$$b_2 = b_1 - L * \frac{dJ}{db}$$

w --> weight
b --> bias
L --> Learning Rate

$\frac{dJ}{dw}$ --> Partial Derivative of cost function with respect to w

$\frac{dJ}{db}$ --> Partial Derivative of cost function with respect to b

Gradients for Lasso Regularization

***if* $(w_j > 0)$:**	***else* $(w_j \leq 0)$:**
$\frac{dJ}{dw} = \frac{-2}{m}\left[\left[\sum_{i=1}^{m} x_j \cdot (y^{(i)} - \hat{y}^{(i)})\right] + \lambda\right]$	$\frac{dJ}{dw} = \frac{-2}{m}\left[\left[\sum_{i=1}^{m} x_j \cdot (y^{(i)} - \hat{y}^{(i)})\right] - \lambda\right]$

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

$$w_2 = w_1 - L * \frac{dJ}{dw}$$

$$b_2 = b_1 - L * \frac{dJ}{db}$$

$$y = w.x + b$$