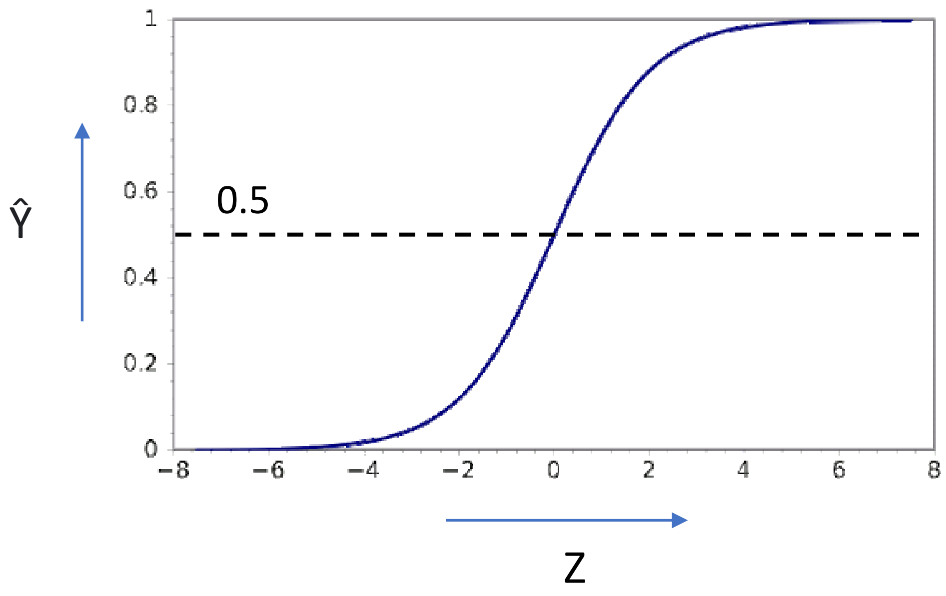
Siddhardhan

Loss Function & Cost Function for Logistic Regression

$J\,(w,b)$

Logistic Regression



$$\hat{Y} = \frac{1}{1 + e^{-Z}}$$

Sigmoid Function

$$Z = w.X + b$$

$\hat{Y}$ - Probability that $(y = 1)$

$\hat{Y} = P(Y=1 \mid X)$

X - input features

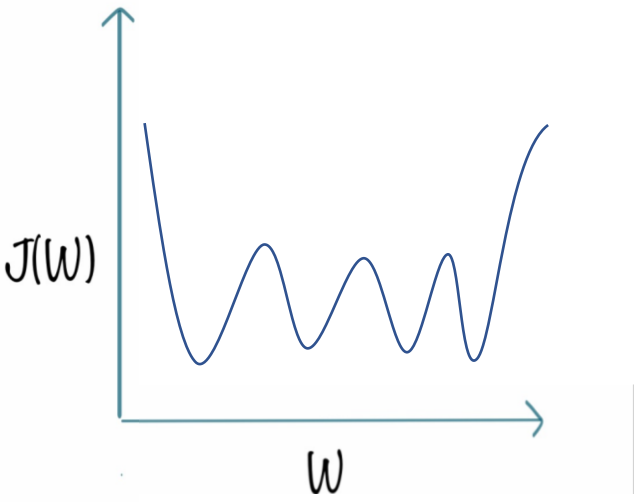
w – weights
(number of weights is equal to the number of input features in a dataset)

b - bias

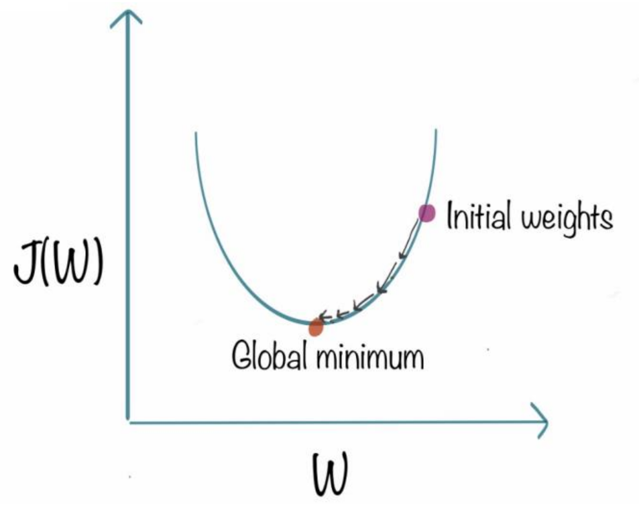
Loss Function

Loss function measures how far an estimated value is from its true value.

$$\textbf{Loss} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$



Gradient Descent
With Local minima



Gradient Descent
With Global minima

Loss Function for Logistic Regression

Binary Cross Entropy Loss Function (or) Log Loss :

$$L (y, \hat{y}) = - (y \log \hat{y} + (1 - y) \log (1 - \hat{y}))$$

When y = 1, $\Rightarrow$ $L (1, \hat{y}) = - (1 \log \hat{y} + (1 - 1) \log (1 - \hat{y}))$ $\Rightarrow$ $L (1, \hat{y}) = - \log \hat{y}$

We always want a smaller Loss Function value, hence, ŷ should be very large,
so that (– log ŷ) will be a large negative number.

When y = 0, $\Rightarrow$ $L (0, \hat{y}) = - (0 \log \hat{y} + (1 - 0) \log (1 - \hat{y}))$ $\Rightarrow$ $L (0, \hat{y}) = - \log (1 - \hat{y})$

We always want a smaller Loss Function value, hence, ŷ should be very small,
so that – log (1 - ŷ) will be a large negative number.

Cost Function for Logistic Regression

Loss function (L) mainly applies for a single training set as compared to the cost function (J) which deals with a penalty for a number of training sets or the complete batch.

$$L(y, \hat{y}) = -(y \log \hat{y} + (1 - y) \log(1 - \hat{y}))$$

$$J(w, b) = \frac{1}{m} \sum_{i=1}^{m} (L(y^{(i)}, \hat{y}^{(i)})) = -\frac{1}{m} \sum_{i=1}^{m} (y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}))$$

('m' denotes the number of data points in the training set)