Week 03

01] Classification

*) Classification is a type of supervised machine learning where the goal is to predict a label or category (class) given input data.

ex :-

-) Email spam detection
-) Fraud transaction detection

Types of classification

1] Binary classification

*) Two possible outcomes
*) 0 (false) is the negative class (absence)
*) 1 (True) is the positive class (presence)

2] Multi - Class Classification

*) more than two categories.

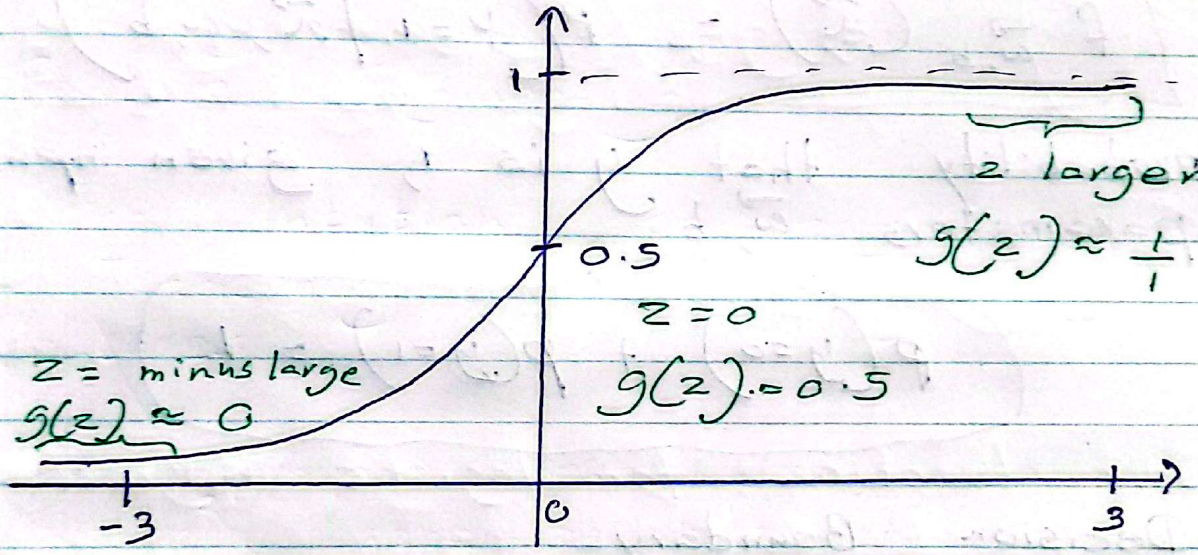
3) Multi - label classification

*) One item can belong to multiple categories (A movie tagged as Comedy and romance)

02] Logistic Regression

*) Despite its name, Logistic Regression is actually used for classification, not regression.

*) It is a supervised learning algorithm used to predict discrete outcomes, like whether something belongs to class 0 or class 1



want outputs between 0 and 1

Sigmoid function
(logistic function)

$$g(z) = \frac{1}{1+e^{-z}} \qquad 0 < g(z) < 1$$

$f_{\vec{w},b}(\vec{x})$, $z = \vec{w} \cdot \vec{x} + b$

$g(z) = \frac{1}{1+e^{-z}} \Rightarrow f_{\vec{w},b}(\vec{x}) = g(\vec{w} \cdot \vec{x} + b)$

$= \frac{1}{1+e^{-(\vec{w} \cdot \vec{x} + b)}}$

Interpretation

$$f_{\vec{w},b}(\vec{x}) = \frac{1}{1+e^{-(\vec{w}\cdot\vec{x}+b)}}$$

"probability" that class is 1

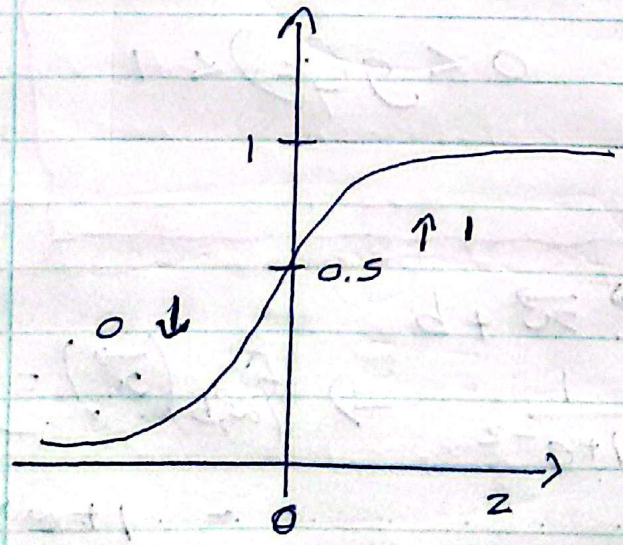
$$f_{\vec{w},b}(\vec{x}) = P(y=1 \mid \vec{x}; \vec{w}, b)$$

* probability that y is 1, given input $\vec{x}$, parameters $\vec{w}, b$

$$P(y=0) + P(y=1) = 1$$

03) Decision Boundary

* A decision boundary is an imaginary line (or surface) that seperates different classes in your data.



$$f_{\vec{w},b}(\vec{x}) = g(\vec{w}\cdot\vec{x}+b) = \frac{1}{1+e^{-(\vec{w}\cdot\vec{x}+b)}}$$

$$= P(y=1 \mid x; \vec{w}, b)$$

IF $f_{\vec{w},b}(\vec{x}) \geq$ 0.5 ?

Threshold

yes: $\hat{y} = 1$

No: $\hat{y} = 0$

when is $f_{\vec{w},b}(\vec{x}) > 0.5$?

$g(z) \geq 0.5$

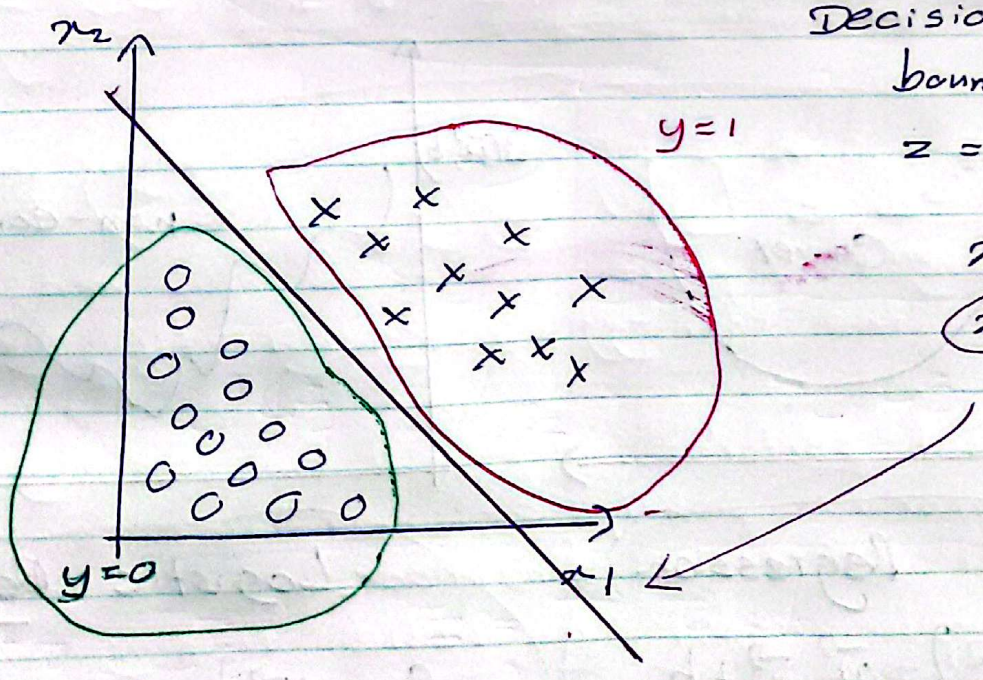
$z \geq 0$

$\vec{w}.\vec{x} + b \geq 0$ $\quad$ $\vec{w}.\vec{x} + b < 0$

$\hat{y} = 1$ $\quad$ $\hat{y} = 0$

$\Rightarrow g(z) = g(w_1 x_1 + w_2 x_2 + b)$

$w_1 = 1, \quad w_2 = 1, \quad b = -3$



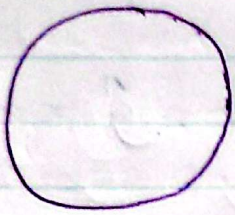
Decision boundary $\Rightarrow$

$z = \vec{w}.\vec{x} + b = 0$

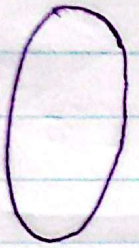
$x_1 + x_2 - 3 = 0$

$x_1 + x_2 = 3$

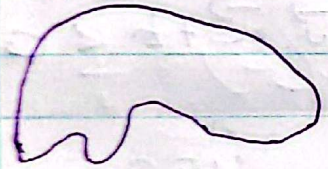
Non linear Decision boundaries



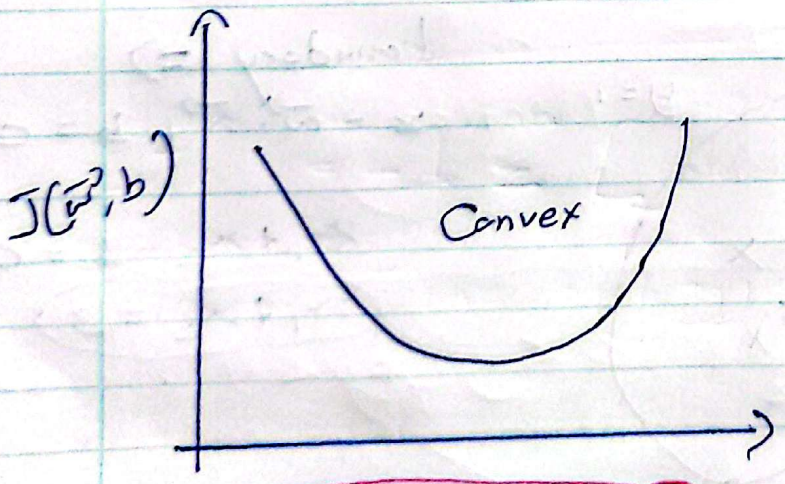
$$\Rightarrow g(w_1 x_1^2 + w_2 x_2^2 + b)$$

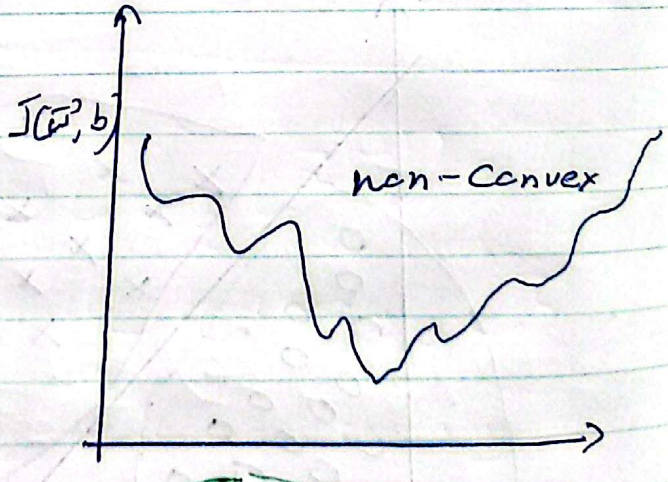


$$\Rightarrow g(z) = g(w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_1 x_2 + w_5 x^2 + w_6 x_1^3 + \dots b)$$



04] Cost function for logistic Regression





Linear Regression

$$f_{\vec{w},b}(\vec{x}) = \vec{w} \cdot \vec{x} + b$$

Logistic Regression

$$f_{\vec{w},b}(\vec{x}) = \frac{1}{1 + e^{-(\vec{w} \cdot \vec{x} + b)}}$$

Cost

$$J(\vec{w}, b) = \frac{1}{m} \sum_{i=1}^{m} \underbrace{\frac{1}{2} \left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{i} \right)^2}_{\text{Loss}}$$

$$L\left(f_{\vec{w},b}(x^{(i)}), y^{(i)}\right)$$

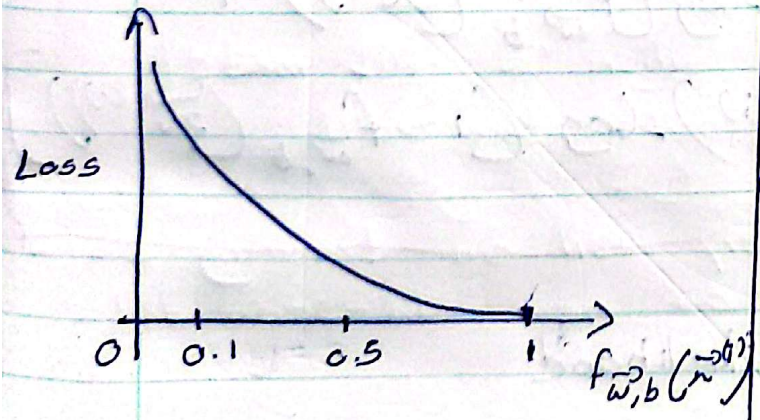
*) Cost function is for model's all training set

*) Loss function is only for one training example set

$$L\left(f_{\vec{w},b}(x^{(i)}, y^{i}\right) = \begin{cases} -\log\left(f_{\vec{w},b}(\vec{x}^{(i)})\right) & \text{if } y^{(i)} = 1 \\ -\log\left(1 - f_{\vec{w},b}(\vec{x}^{(i)})\right) & \text{if } y^{(i)} = 0 \end{cases}$$

find w, b that minimize cost J

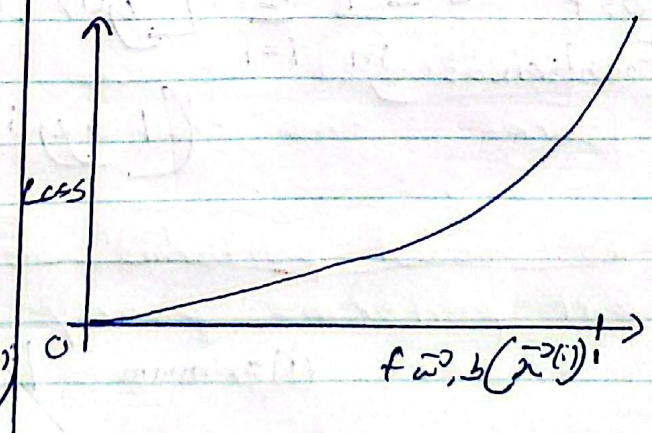
if $y^{(i)} = 1$



$f_{\vec{w},b}(x^{(i)}) \to 1$ then loss $= 0$

$f_{\vec{w},b}(x^{(i)} \to 0$ then loss $= \infty$

if $y^{(i)} = 0$



$f_{\vec{w},b}(x^{(i)}) \to 1$ then loss $= \infty$

$f_{\vec{w},b}(x^{(i)} \to 0$ then loss $= 0$

*) The further prediction $f_{\vec{w},b}(\vec{x}^{(i)})$ is from target $y^{(i)}$, the higher the loss

Simplified Cost function for Logistic Regression:

$$L(f_{\vec{w},b}(\vec{x}^{(i)}, y^{(i)}) = -y^{(i)} \log(f_{\vec{w},b}(\vec{x}^{(i)})) - (1-y^{(i)}) \log(1 - f_{\vec{w},b}(\vec{x}^{(i)}))$$

Log loss (Convex)

$$J(\vec{w},b) = \frac{1}{m} \sum_{i=1}^{m} [L(f_{\vec{w},b}(\vec{x}^{(i)}, y^{(i)})]$$

Cost Function $= -\frac{1}{m} \sum_{i=1}^{m} [y^{(i)} \log(f_{\vec{w},b}(\vec{x}^{(i)})) + (1-y^{(i)}) \log(1 - f_{\vec{w},b}(\vec{x}^{(i)}))]$

Maximum likelihood

05) Gradient descent for logistic Regression

$$J(\vec{w}, b) = \frac{-1}{m} \sum_{i=1}^{m} \left[y^{(i)} \log\left(\vec{x}^{(i)}\right) + \left(1 - y^{(i)}\right) \log \left(1 - f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) \right]$$

repeat

{

$$w_j = w_j - \alpha \underbrace{\frac{\partial}{\partial w_j} J(\vec{w}, b)}$$

$$\frac{1}{m} \sum_{i=1}^{m} \left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right) - y^{(i)}\right) x_j^{(i)}$$

$$b = b - \alpha \underbrace{\frac{\partial}{\partial b} J(\vec{w}\, b)}$$

$$\frac{1}{m} \sum_{i=1}^{m} \left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right) - y^{(i)}\right)$$

} simultaneous updates

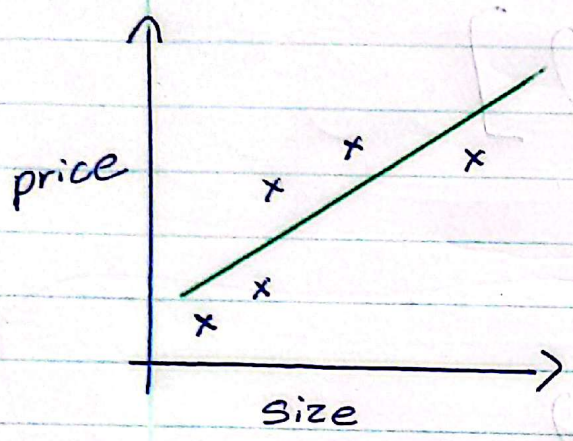
*) As gradient descent for both linear and logistic looks same it is the $f_{\vec{w},b}(\vec{x})$ that changes.

linear regression = $\vec{w} \cdot \vec{x} + b$

logistic regression = $\frac{1}{1 + e^{-(\vec{w} \cdot \vec{x} + b)}}$

06] The problem overfitting

Regression Example

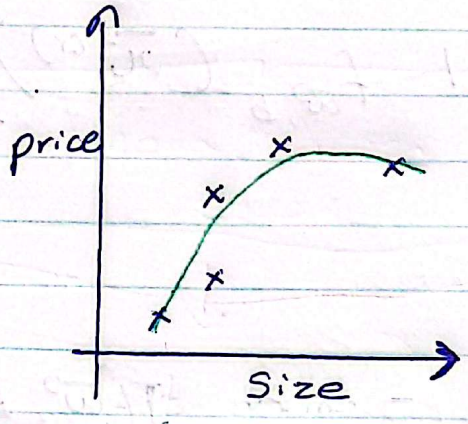


$w_1 x + b$

Underfit

*) Does not fit the training set well

High bias

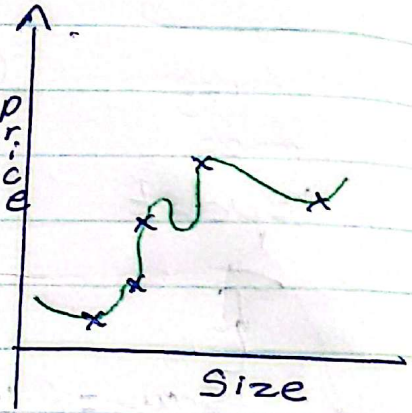


$w_1 x + w_2 x^2 + b$

Just Right

*) Fits training set pretty well

Genarlization



$w_1 x + w_2 x^2 + w_3 x^3 + w_6 x^6 + b$

Overfit

*) Fits the trai -ning set extremly well

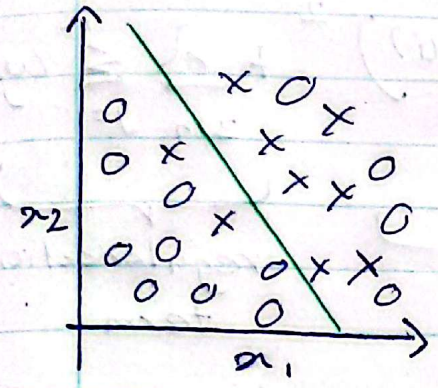
High Variance

*) Overfitting happens when a ML model learns the training data too well.

*) The model performs very well on training data But performs poorly on new un-seen data

*) Too, compler model, Too little training data, No regularization, Too many features can cause overfitting

Classification Example

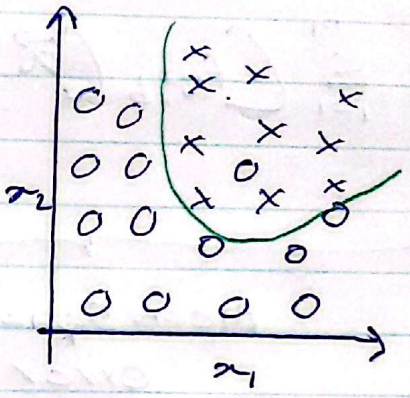


$z = w_1 x_1 + w_2 x_2 + b$

$f_{\vec{w},b}(\vec{x}) = g(z)$

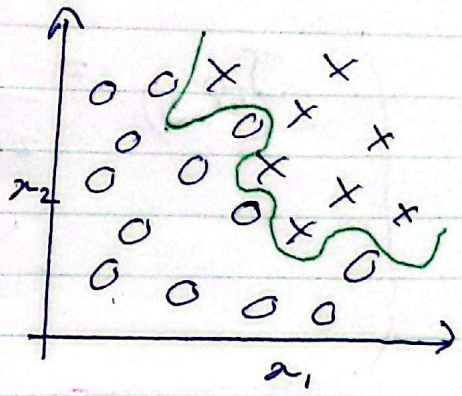
g is the sigmoid function

Underfit
High bias



$z = w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_2^2 + w_5 x_1 x_2 + b$

Just right



$z = w_1 x_1 + w_2 x_2 + w_3 x_1^2 x_2 + w_4 x_1^2 x_2^2 + w_5 x_1^2 x_2^3 + w_6 x_1^3 x_2 + \dots b$

Overfit

How to prevent Overfitting

1) Use more training data - More examples help the model generalize

2) Regularization - penalizes large weights to prevent 'extreme fitting

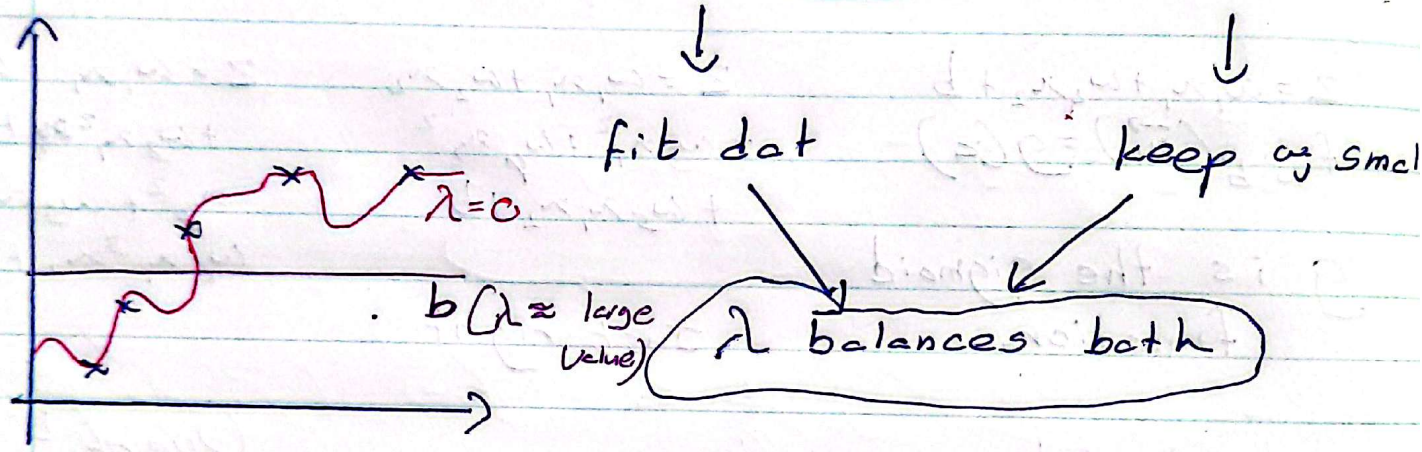
3) Feature selection - Remove irrelevant features that cause noise

07) Cost function with regularization

$$J(\vec{w}, b) = \frac{1}{2m} \sum_{i=1}^{m} \left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)^2 + \frac{\lambda}{2m} \sum_{j=1}^{n} w_j^2$$

mean squared error

regularization term



*) If $\lambda = 0$ overfit
If λ enormus (like 10^{10}) underfit

Need to chose a λ in between

08) Regularized linear Regression

$$J(\vec{w}, b) = \frac{1}{2m} \sum_{i=1}^{m} \left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)^2 + \frac{\lambda}{2m} \sum_{j=1}^{n} w_j^2$$

Gradient descent

repeat {

$$w_j = w_j - \alpha \frac{\partial}{\partial w_j} J(\vec{w}, b)$$

$$\frac{\partial}{\partial w_j} J(\vec{w}, b) = \frac{1}{m} \sum_{i=1}^{m} \left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)} \right) x_j^{(i)} + \frac{\lambda}{m} w_j$$

$$b = b - \alpha \underbrace{\frac{\partial}{\partial b} J(\vec{w}, b)}$$

$$= \frac{1}{m} \sum_{i=1}^{m} \left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)} \right)$$

Don't have to regularize b

3 Simultaneous update

Implementing Gradient descent

$$w_j = w_j - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} \left[f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)} \right] x_j^{(i)} + \frac{\lambda}{m} w_j \right]$$

$$b = b - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)} \right)$$

*) For logistic regression it looks same but $f(\vec{w})$ changes $\left(\frac{1}{1+e^{z}} \right)$

$$w_j = w_j \underbrace{\left(1 - \alpha \frac{\lambda}{m} \right)}_{\text{new}} - \underbrace{\alpha \frac{1}{m} \sum_{i=1}^{m} \left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)} \right) x_j}_{\text{Original part}}$$