Regression

used when output is continuous

with one variable

$\hat{y} = wx + b$

Cost function

number of samples

$J(w,b) = \frac{1}{2m} \sum_{i=1}^{m} (\hat{y}^i - y^i)^2$

$\frac{1}{2m} \sum_{i=1}^{m} (wx^i + b - y^i)^2$

Squared error function

Gradient descent

$w = w - \alpha \boxed{\frac{d}{dw} J(w,b)}$

$b = b - \alpha \boxed{\frac{d}{db} J(w,b)}$

Batch gradient descent.
all training example

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

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

with multi-li more than one var

Previous

Parameters

$w_1, \ldots, w_n$
b

Model

$\hat{y}_{\vec{w},b}(\vec{x}) = w_1 x_1 + \cdots + w_n x_n + b$

Cost function

$J(w_1, \ldots w_n, b)$

Gradient descent

$w_i = w_i - \alpha \frac{d}{dw_i} J(w_1, \ldots, w_n, b)$

$b = b - \alpha \frac{d}{db_i} J(w_1, \ldots, w_n, b)$

Vector notation

numpy

$\vec{w} = [w_1 \ldots w_n]$
b

$\hat{y}_{\vec{w},b}(\vec{x}) = \vec{w} \cdot \vec{x} + b$

$J(\vec{w}, b)$

Features

Normalization

if we have more than one features and if features values have gap like $300 \leq x \leq 2000$ and $0 \leq y \leq 5$

so it is better to normalize to have better assumption

Mean normalization

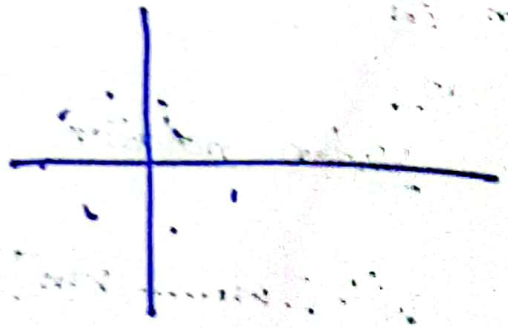
$$x = \frac{x - \mu_1}{2000 - 300}$$ average

$$y = \frac{y - \mu_2}{5 - 0}$$

which in our case will be 600

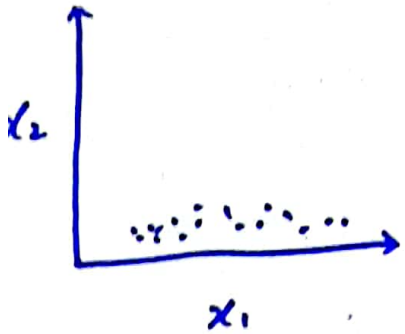
$-0.18 \leq x \leq 0.82$

$-0.46 \leq y \leq 0.54$



after normalizing values will be centred around 0

Z score normalization



$x_1 = \frac{x_1 - \mu_1}{\sigma_1}$ — μ_1 — mean

σ_1 — standard deviation.

Purpose of feature scaling

to have value $-1 \leq x \leq 1$

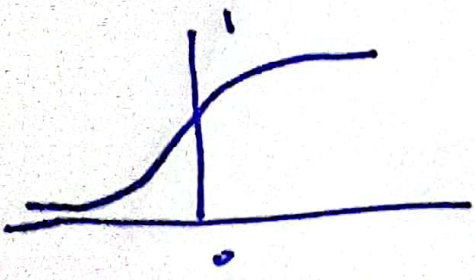
Logistic regression

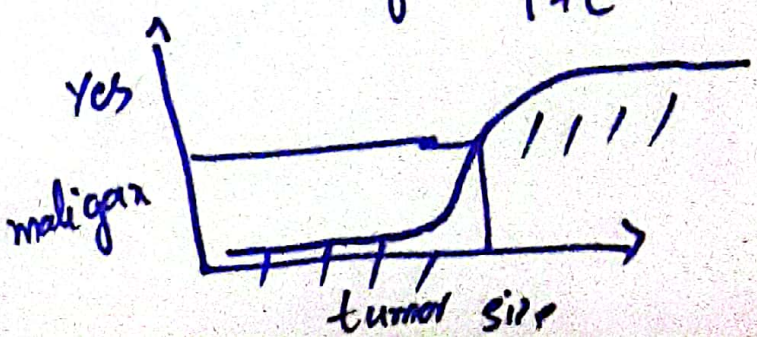
- Classification

Sigmoid function

$0 < g < 1$

$g = \frac{1}{1+e^{-x}}$





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

$$z = \vec{w} \cdot \vec{x} + b$$

$$\downarrow$$

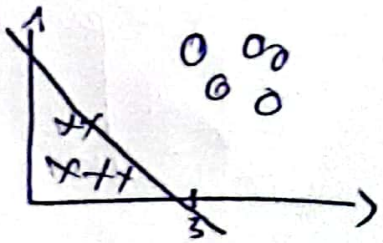
$$g(z) = \frac{1}{1+e^{-z}}$$

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

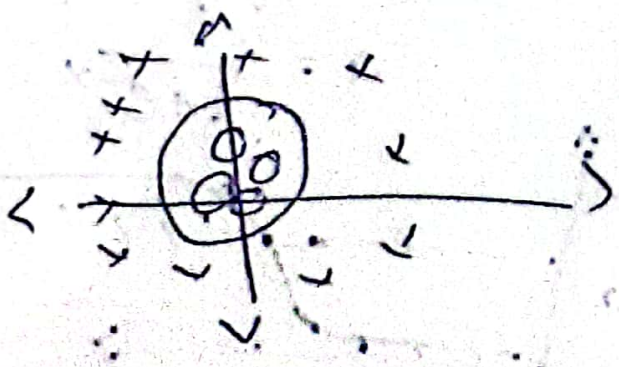
$$= \frac{1}{1+e^{-(\vec{w} \cdot \vec{x} + b)}}$$

→ logistic regression

Decision boundary



$x_1 + x_2 = 3$



$x_1^2 + x_2^2 \leq 3$

Cost function

x_1 Tumor size	x_n age	malignant y
10	52	1
2	73	0
5	55	0
12	49	1

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

How to choose $\vec{w}$ and b

- Squared error cost

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

non convex

logistic

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

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

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

$f_{\vec{w},b}(\vec{x}) \longrightarrow 1$ then loss 0

$f_{\vec{w},b}(\vec{x}) \rightarrow 0$ then loss ∞

(if real value y is 1)

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

Simplified loss function

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

So now cost:-

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

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

Gradient descent

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

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

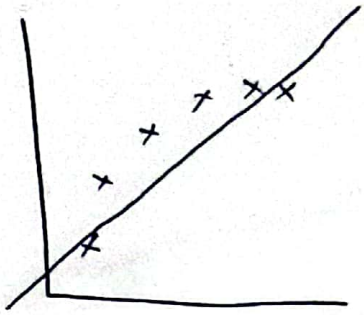
$$b = b - \alpha \frac{d}{db} J(\vec{w},b)$$

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

Overfitting

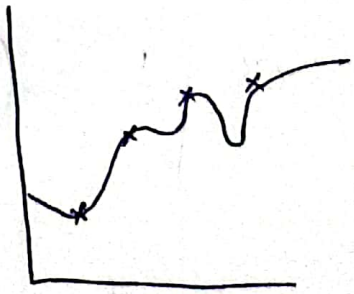
underfit

- doesnot fit training set well.
- high bias



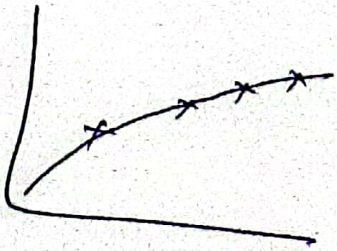
overfit

- Fits training set extremely well
- high variance



generalized

fits training set well



Regularization

if model overfit

$$J(w,b) = \underbrace{\frac{1}{2m}\sum_{i=1}^{m}\left(f_{w,b}(\vec{x}^{(i)}) - y^{(i)}\right)^2}_{\text{mean squared error}} + \underbrace{\frac{\lambda}{2m}\sum_{j=1}^{n} w_j^2}_{\text{regularization term}}$$

keep w small

λ balance both

data fit

gradient descent

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

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

update cost

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