LINEAR REGRESSION

Supervised Learning

$X \rightarrow Y$ mapping

$n \rightarrow$ Number of x, y exs in training set

$d \rightarrow x \in \mathbb{R}^d$ dimension of input

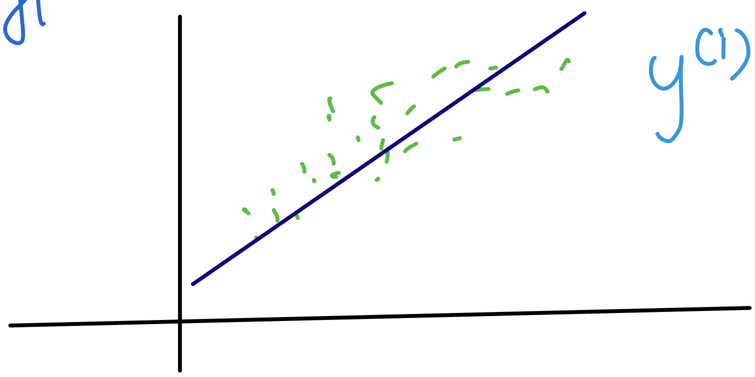
$x^{(i)} \rightarrow i^{th}$ input example

$y^{(i)} \rightarrow i^{th}$ example output/label/ground truth

$(x^{(i)}, y^{(i)}) \rightarrow i^{th}$ example

Learn hypothesis $\rightarrow h(x) \approx Y$ (Regression)

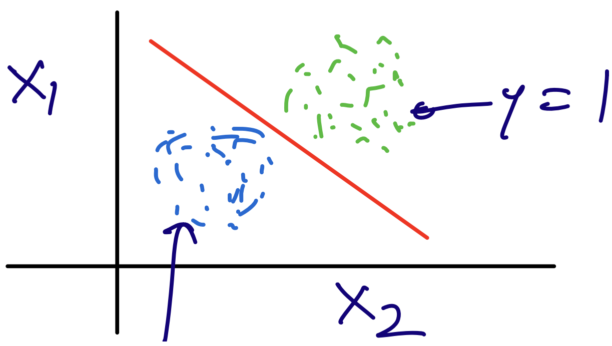
X	Y
⋮	⋮



$y^{(i)} \in \mathbb{R}$

Classification

x_1	x_2	y
		0
		1



$y^{(i)} \in \{0, 1\}$ $y = 0$

LINEAR REGRESSION

$x \in \mathbb{R}^d$ $y \in \mathbb{R}$, $n \to$ such examples

$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 \ldots \theta_d x_d$

$\theta \in \mathbb{R}^{d+1}$

$$h_\theta(x) = \left(\sum_{i=1}^{d} \theta_i x_i\right) + \theta_0$$

$\boxed{x_0 = 1}$ ← Intercept term

$$h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 \ldots, \theta_d x_d$$

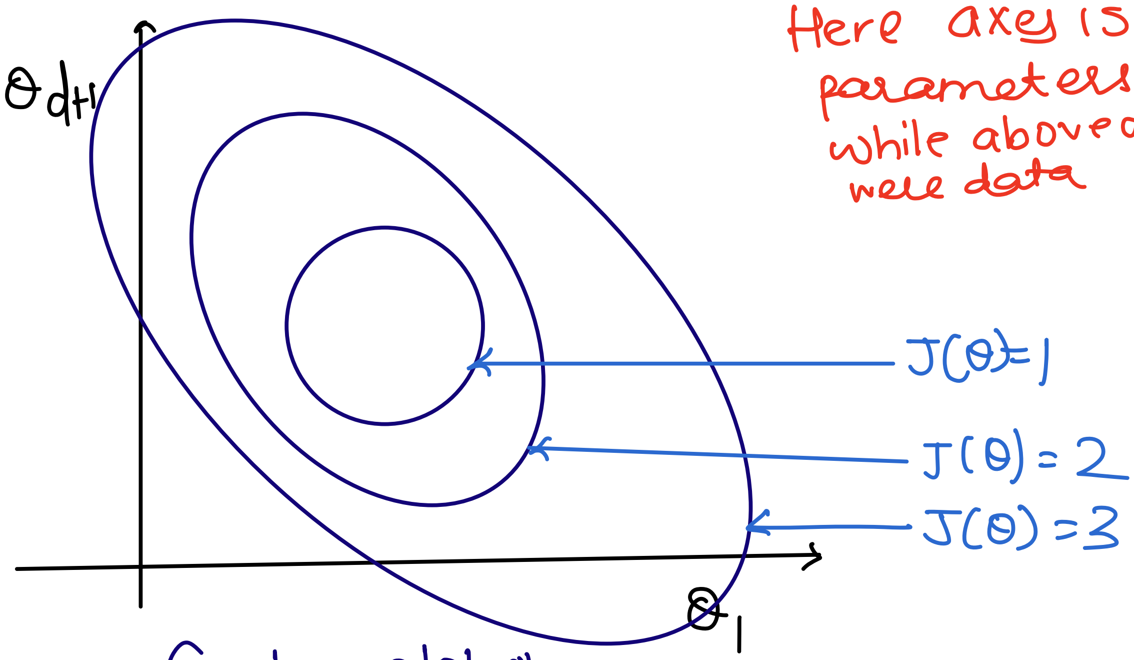
$$= \theta^T x \quad \left(\sum_{i=0}^{d} \theta_i x_i\right)$$

Cost / Loss Function

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{n} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

$$\hat{\theta} \in \arg\min_\theta J(\theta)$$ (Multiple values of θ to minimize)

$$= \arg\min_\theta \sum_{i=1}^{n} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$



Here axes is the parameters while above axes were data

Contour plot of $J(\theta)$

This is Bowl Shaped

[Dome Shaped if $J(\theta)$ is in opp order]

$$\theta^{(0)} := \text{Initialization}$$

$$\theta_j^{(1)} = \theta_j^{(0)} - \alpha \frac{\partial}{\partial \theta_j} J(\theta^{(0)})$$

Repeat till convergence

$$\boxed{\theta^{(1)} = \theta^{(0)} - \alpha \nabla_\theta J(\theta^{(0)})}$$

If cost function is convex, no matter how you initialize you will reach minima

Checks for Convergence

$$\|\nabla_\theta J(\theta^{(t)})\|, \quad \|\theta^{(t)} - \theta^{(t-1)}\|, \quad \|J(\theta^{(t)}) - J(\theta^{(t-1)})\|$$

$$\theta^{(t+1)} = \theta^{(t)} - \alpha \nabla_\theta [J(\theta^{(t)})]$$

$$= \theta^{(t)} - \alpha \nabla_\theta \left[\frac{1}{2} \sum_{i=1}^{n} (h_\theta(x^{(i)}) - y^{(i)})^2 \right]$$

$$= \theta^{(t)} - \alpha \nabla_\theta \left[\frac{1}{2} \sum_{i=1}^{n} (\theta^T x^{(i)} - y^{(i)})^2 \right]$$ ith example

iteration

$$= \underbrace{\theta^{(t)}}_{\mathbb{R}^d} - \alpha \underbrace{\sum_{i=1}^{n} (\theta^T x^{(i)} - y^{(i)}) \cdot x^{(i)}}_{\mathbb{R}^d}$$

For gradient descent, for a small step progress, iterate over whole eg. [computationally expensive]

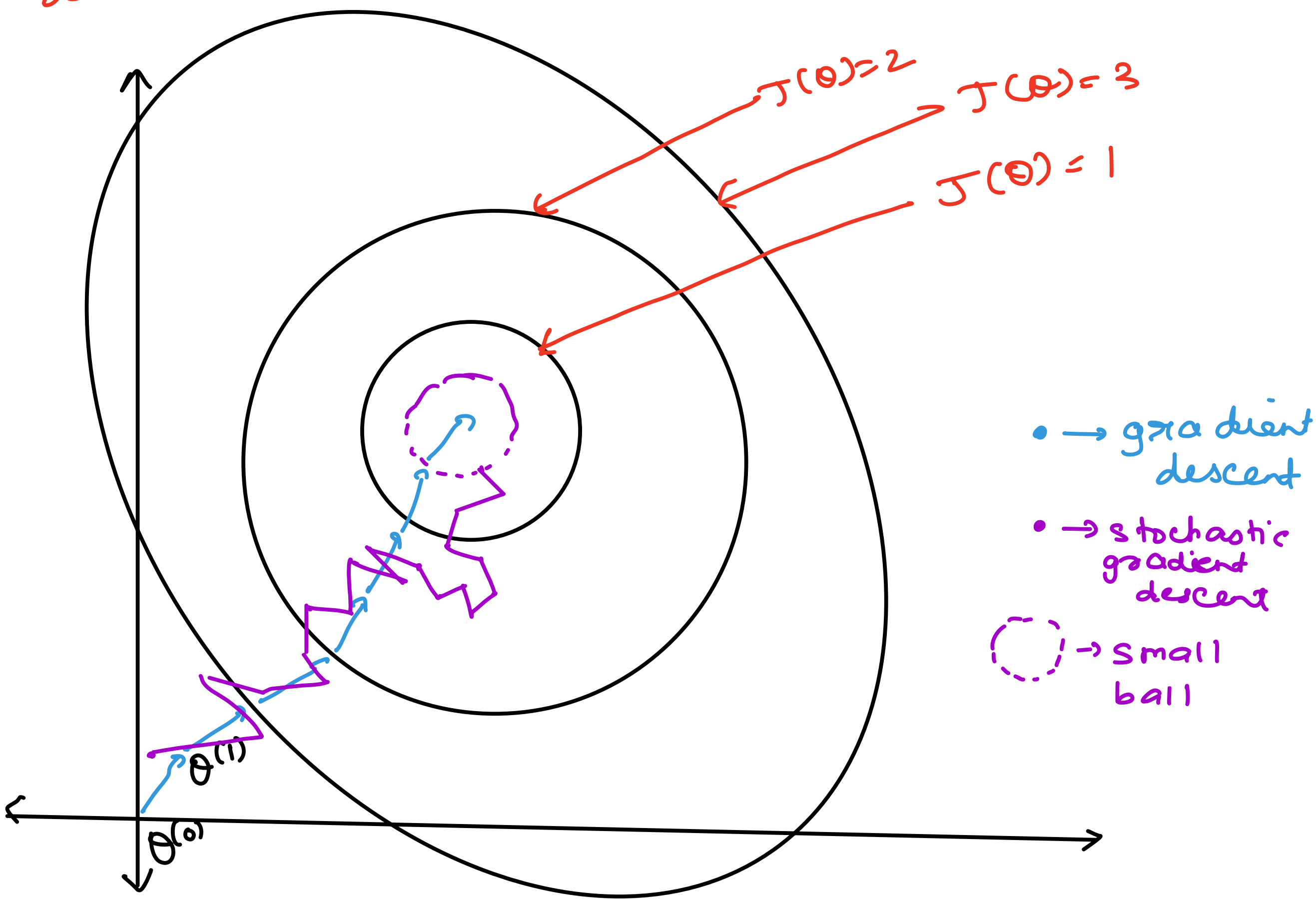
Stochastic Gradient Descent:-

$$\theta^{(t+1)} = \theta^{(t)} - \alpha \nabla_\theta \tilde{J}(\theta)$$

$$\tilde{J}(\theta) = \frac{1}{2}\left(\theta^T x^{(k)} - y^{(k)}\right)^2$$

$k \rightarrow$ uniformly randomly sampled from training set

We take one eg. & calculate $J(\theta)$, then take that step accordingly



When you use S.G.D., you reach a region around true minima., and all further updates will be confined to that ball whose radius will be a function of step size α.

S.G.D. takes more steps but cost of each step is so low it is worth it

D.L. → non-convex cost function

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{n}\left(\theta^T x^{(i)} - y^{(i)}\right)^2$$

Design Matrix

$$X = \begin{array}{c|c} \begin{bmatrix} - x^{(1)} - \\ \vdots \\ - x^{(i)} - \\ \vdots \\ - x^{(n)} - \end{bmatrix} & \updownarrow n \\ \overleftrightarrow{\;d\;} & \end{array} \qquad \vec{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

$$\theta = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_d \end{bmatrix}$$

dot product interpretation

$$\underset{\mathbb{R}^{n \times d}}{X}\;\underset{\mathbb{R}^{d}}{\theta} - \underset{\mathbb{R}^{n}}{y} = \begin{bmatrix} x^{(1)T}\theta \\ \vdots \\ x^{(i)T}\theta \\ \vdots \\ x^{(n)T}\theta \end{bmatrix} - \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(n)} \end{bmatrix}$$

$= \mathbb{R}^n - \mathbb{R}^n$

$\mathbb{R}^n$

$$= \begin{bmatrix} x^{(1)T}\theta - y^{(1)} \\ \vdots \\ x^{(i)T}\theta - y^{(i)} \\ \vdots \\ x^{(n)T}\theta - y^{(n)} \end{bmatrix}$$

$$J(\theta) = \frac{1}{2}(X\theta - y)^T(X\theta - y)$$ → Dot product & then sum up

$$\nabla_\theta J(\theta) = \frac{1}{2}\nabla_\theta\left((X\theta - y)^T(X\theta - y)\right)$$

$$= \frac{1}{2}\nabla_\theta\left[\left((X\theta)^T - y^T\right)(X\theta - y)\right]$$

$$= \frac{1}{2}\nabla_\theta\left[\left(\theta^T X^T - y^T\right)(X\theta - y)\right]$$

$$= \frac{1}{2}\nabla_\theta\left(\theta^T X^T X\theta - \theta^T X^T y - y^T X\theta + \cancel{y y^T}\right)$$

$$= \frac{1}{2}\nabla_\theta\left(\theta^T(X^TX)\theta - 2\theta^T(X^Ty)\right)$$

$$= \frac{1}{2}\left[2(X^TX)\theta - 2X^Ty\right]$$

$$= X^TX\theta - X^Ty$$

$$X^TX\theta = X^Ty$$

$$\boxed{\hat{\theta} = (X^TX)^{-1}X^Ty}$$

Probabilistic Interpretation

$y^{(i)} = \theta^T x^{(i)} + \varepsilon^{(i)}$ → Assumption

$\varepsilon^{(i)} \sim N(0, \sigma^2)$

$$\varepsilon^{(i)} = \underbrace{y^{(i)} - \theta^T x^{(i)}} \sim N(0, \sigma^2)$$

$$P(y^{(i)} \mid x^{(i)}; \theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)$$

$$y^{(i)} \sim N(\theta^T x^{(i)}, \sigma^2)$$

Data	Parameter
X, Y	μ, σ^2
	θ, σ^2

$$\log L(\theta) = \log \prod_{i=1}^{n} p(y^{(i)} \mid x^{(i)}; \theta)$$

$$\ell(\theta) = \sum_{i=1}^{n} \log p(y^{(i)} \mid x^{(i)}; \theta)$$

$$\ell(\theta) = \sum_{i=1}^{n} \log\left[\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-1}{2} \frac{(y^{(i)} - \theta^T x^{(i)})^2}{\sigma^2}\right)\right]$$

$$= \sum_{i=1}^{n} \left(K - \frac{1}{2\sigma^2} (y^{(i)} - \theta^T x^{(i)})^2\right)$$

$$= nK - \frac{1}{2\sigma^2} \left[\sum_{i=1}^{n} \frac{1}{2} (y^{(i)} - \theta^T x^{(i)})^2\right]$$

Squared
Error

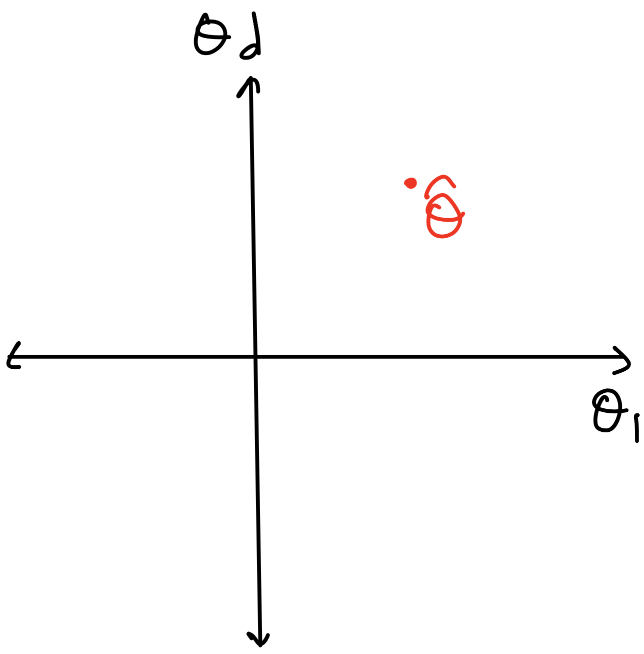
So by maximizing $\ell(\theta)$, we minimize squared error

Hence, noise is gaussian

(Note our final value of θ doesn't depend on value of σ^2)

$$\arg\max_{\theta} \ell(\theta) = \arg\min_{\theta} J(\theta)$$

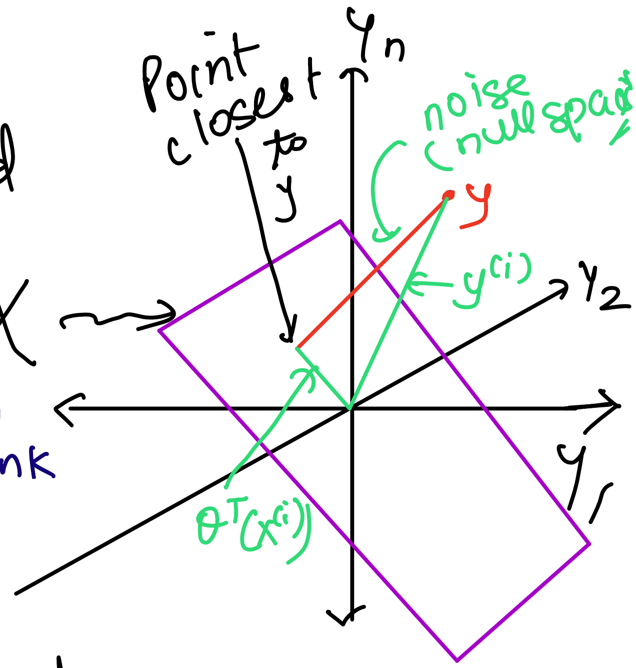
Solve: $X\theta \approx Y$



$n \times d$

X

Full Rank



$$n >> d$$

$$X\hat{\theta} = X(X^TX)^{-1}X^Ty$$

$$\hat{\theta} = (X^TX)^{-1}X^Ty$$

Observed $\bar{y}$ never lies in subspace $(n >> d)$

$X\hat{\theta} \simeq Y$ (never $=$)

$X^T X$ is not invertible when

→ No. of linearly independent egs < No. of feature

→ Features not linearly indepen-dent

(Solⁿ: Regularisation)

Locally Weighted Linear Regression

→ Assuming there is sufficient training data, makes selection of features less critical

Normal L.R.	Locally weighed L.R.
(1) Fit θ to minimize $\sum_i \left(y^{(i)} - \theta^T x^{(i)}\right)^2$	① Fit θ to minimize $\sum_i \left(y^{(i)} - \theta^T x^{(i)}\right)^2 \cdot w^{(i)}$
② Output $\theta^T x$	② Output $\theta^T x$
	(Here $w^{(i)}$ is non-neg. value called weight)

If weight is large, we will try very hard to minimize $(y^{(i)} - \theta^T x)^2$ but if weight is small, we might ignore it.

Fairly standard choice

$$w^{(i)} = \exp\left(-\frac{(x^{(i)} - x)^2}{2\tau^2}\right)$$

$x \to$ point at which evaluation going on.

So, for point at which $|x^{(i)} - x|$ is small, $w^{(i)}$ is close to 1

If $|x^{(i)} - x|$ is large, $w^{(i)}$ is very small.

Thus θ is chosen, giving much higher weights to (error on) points closer to query point x

This is a Non Parametric Model since we need to keep entire training set around. Non Parametric means the amount of stuff needed to store to represent hypothesis h grows linearly with size of training set

Normal LR is Parametric. We once calculate θ by fitting our model, then we don't need to store it while inference.