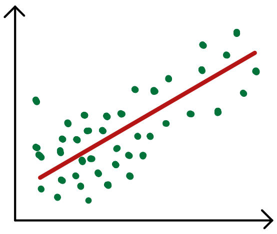
LINEAR REGRESSION

Hypothesis $h(x) = \sum_{j=0}^{n} \theta_j x_j$ where $x_0 = 1$

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

$\underset{\theta}{\text{minimize}}\ J(\theta)$



θ = parameters
m = no of training examples
x = inputs/features
y = output/target
(x,y) = training example
$(x^{(i)}, y^{(i)})$ = i^{th} training example
$J(\theta)$ = cost function
α = learning rate
n = number of features

Gradient Descent

Batch Gradient Descent

Start with some θ $(\theta = \vec{0})$
Keep reducing θ to reduce $J(\theta)$

for $j = 0$ to n {

$$\theta_j := \theta_j - \alpha \frac{\partial J(\theta)}{\partial \theta_j}$$

$$\theta_j := \theta_j - \alpha \sum_{i=1}^{m}\left(h(x^{(i)}) - y^{(i)}\right) x_j^{(i)}$$

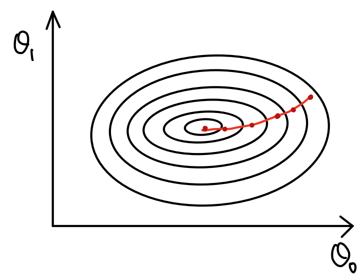
}

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{\partial}{\partial \theta_j} \frac{1}{2} \sum_{i=1}^{m}\left(h(x^{(i)}) - y^{(i)}\right)^2$$

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{2}{2} \sum_{i=1}^{m}\left(h(x^{(i)}) - y^{(i)}\right) \frac{\partial}{\partial \theta_j} \sum_{j=0}^{n} \theta_j x_j^{(i)} - y^{(i)}$$

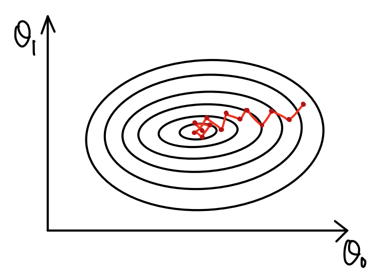
$$\frac{\partial J(\theta)}{\partial \theta_j} = \sum_{i=1}^{m}\left(h(x^{(i)}) - y^{(i)}\right) x_j^{(i)}$$

In batch gradient descent the gradient is evaluated on the entire data set and parameters are updated after each epoch.
Batch gradient descent converges to local minima.



Stochastic Gradient Descent

Repeat {
for $i = 1$ to m {
for $j = 0$ to n $\{\theta_j := \theta_j - \alpha\left(h(x^{(i)}) - y^{(i)}\right) x_j^{(i)}\}$
}
}



In stochastic grdient descent the gradient is evaluated on each data point and parameters are updated after each data point. It does not converge on the minima but oscillates near the local minima.

Normal Equation

Normal Equation: Normal equation helps us find the optimal value of θ and jump to the global minima in a single step. Normal equation works only for linear regression.

$J(\theta)$ = Cost function mapping parameters to real numbers
$\nabla_\theta J(\theta)$ = Derivitive of $J(\theta)$ wrt θ

$$= \begin{bmatrix} \frac{\partial J}{\partial \theta_0} \\ \vdots \\ \frac{\partial J}{\partial \theta_n} \end{bmatrix}$$

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

$J(\theta)$ for Linear regression has only global minima and no global maxima as $J(\theta)$ is a paraboloid for linear regression

$\nabla_\theta J(\theta) = \vec{0}$ gives the global minima

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(h(x^{(i)}) - y^{(i)})^2$$

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

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

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

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

$$= X^T X\theta - X^T y = 0$$

$$X^T X\theta = X^T y$$

Normal equation $\theta = (X^T X)^{-1} X^T y$

$$X\theta = \begin{bmatrix} (x^{(1)})^T \\ \vdots \\ (x^{(m)})^T \end{bmatrix} \times \begin{bmatrix} \theta_0 \\ \vdots \\ \theta_n \end{bmatrix} = \begin{bmatrix} x^{(1)T}\theta \\ \vdots \\ x^{(m)T}\theta \end{bmatrix} = \begin{bmatrix} h(x^{(1)}) \\ \\ h(x^{(m)}) \end{bmatrix}$$

X = design matrix
θ = parameters
Y = output matrix

$$\bar{y} = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{bmatrix} \qquad x^{(i)} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

1st sample of training set