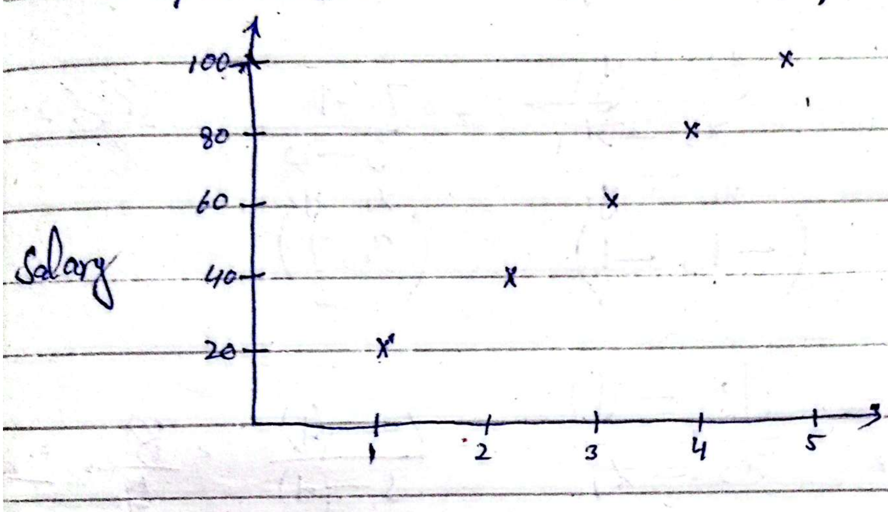
Linear Regression with Gradient Descent Algorithm

Given Dataset:

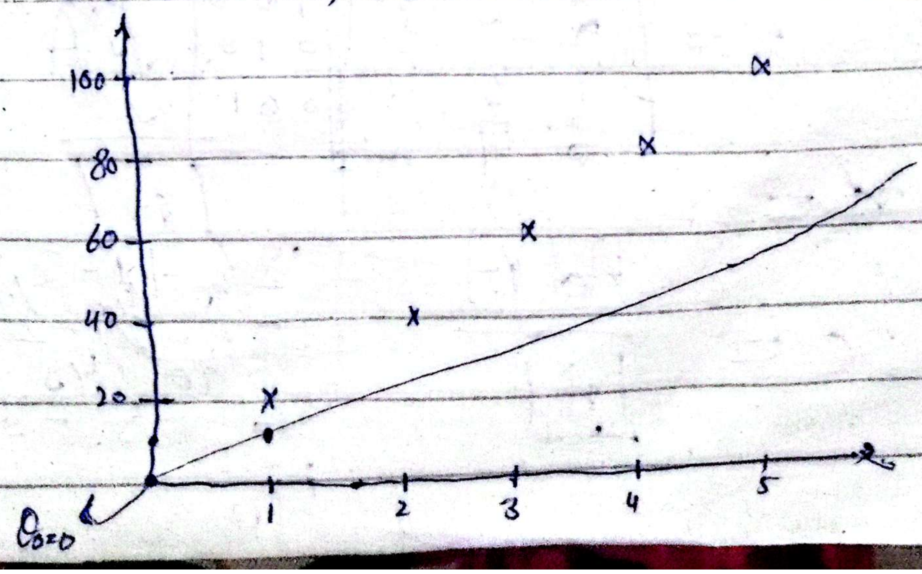
Input ← Experience	Salary → output
1	20
2	40
3	60
4	80
5	100

Step 1: Draw Dataset In Graphe



Step 2: Draw a line / hypothisis model with assume θ_0, θ_1

$\theta_0 = 0$, $\theta_1 = 10$.



Step 3: Find the pridicted Value with formula $\hat{Y} = \theta_0 + \theta_1 X_i$

$$\underbrace{\hat{Y}}_{\text{Pridict}} = \underbrace{\theta_0}_{\text{assume Given}} + \underbrace{\theta_1}_{\text{assume Given}} \underbrace{X_i}_{\text{Input feature}}$$

Input Experience	output Salary	Pridicted Salary
1	20	10
2	40	20
3	60	30
4	80	40
5	100	50

$\theta_0 = 0$, $\theta_1 = 10$

(i) $\hat{Y} = 0 + 10(1) = 10$

(ii) $\hat{Y} = 0 + 10(2) = 20$

(iii) $\hat{Y} = 0 + 10(3) = 30$

(iv) $\hat{Y} = 0 + 10(4) = 40$

(v) $\hat{Y} = 0 + 10(5) = 50$

Step 4: Find the Error with

$$MSE = \frac{1}{\underbrace{\cancel{2}m}_{\text{Total Sample}}} \sum_{i=1}^{m} \left(\underbrace{\hat{Y}}_{\text{Pridict}} - \underbrace{Y}_{\text{actual value}} \right)^2$$

(کہیں $\frac{1}{2m}$ استعمال ہوتا ہے)

$$MSE = \frac{1}{\cancel{2}m} \left[(10-20)^2 + (20-40)^2 + (30-60)^2 + (40-80)^2 + (50-100)^2 \right]$$

$$MSE = \frac{1}{\cancel{2}m} \left[5500 \right] \Rightarrow \boxed{m = 5}$$

$$MSE = \frac{5500}{5} = 1100$$

اگر MSE کی ویلیو زیادہ ہوگی تو Model اتنا ہی خراب ہوگا۔ اگر MSE کم ہوگی تو Model اتنا ہی اچھا ہوگا۔

Step 5: update θ_0, θ_1 Value with the help of Gradient descent Algorithm with formulla

$$\theta_i = \theta_i - \alpha \frac{d}{d\theta_i} \text{Cost function}$$

1) Calculat derivative of Cost function with respect to θ_0

$$\theta_0 = \theta_0 - \alpha \frac{d}{d\theta_0} \frac{1}{m} \sum_{i=1}^{m} (\hat{Y} - Y)^2$$

$$\theta_0 = \theta_0 - \alpha \left(\frac{1}{m} 2 \sum_{i=1}^{m} (\hat{Y} - Y) \frac{d}{d\hat{Y}} (\hat{Y} - Y) \right)$$

$$= \theta_0 - \alpha \left(\frac{2}{m} \sum_{i=1}^{m} (\hat{Y} - Y) * 1 \right.$$

$$\theta_0 = \boxed{\theta_0 - \alpha \left(\frac{2}{m} \sum_{i=1}^{m} (\hat{Y} - Y) \right)}$$

2) Calculate derivative of Cost function with respect to θ_1

$$\theta_1 = \theta_1 - \alpha \frac{d}{d\theta_1} \left(\frac{1}{m} \sum_{i=1}^{m} (\hat{Y} - Y)^2 \right)$$

$$\theta_1 = \theta_1 - \alpha \left(\frac{2}{m} \sum_{i=1}^{m} (\hat{Y} - Y) \cdot \underbrace{X_i}_{\text{Input feature Sample}} \right)$$

$$\boxed{\theta_1 = \theta_1 - \alpha \left(\frac{2}{m} \sum_{i=1}^{m} (\hat{Y} - Y) X_i \right)}$$

$\theta_0 = 0$, $\theta_1 = 10$, assum $\alpha = 0.01$

1) $\theta_0 = \theta_0 - \alpha\left(\frac{2}{m}\sum_{i=1}^{m}(\hat{Y} - Y)\right)$

$\theta_0 = 0 - 0.01\left(\frac{2}{5}\left[(10-20)+(20-40)+(30-60)+(40-80)+(50-100)\right]\right)$

$\theta_0 = 0 - 0.01\left(\frac{2}{5}(-150)\right)$

$\theta_0 = 0 - 0.01\left(-\frac{300}{5}\right)$

$\theta_0 = 0 - 0.01(-60)$

$\boxed{\theta_0 = 0.6}$

2) $\theta_1 = \theta_1 - \alpha\left(\frac{2}{m}\sum_{i=1}^{m}(\hat{Y} - Y)X_i\right)$

Input sample

$\theta_1 = 10 - 0.01\left(\frac{2}{5}\left[(10-20)*1+(20-40)*2+(30-60)*3+(40-80)*4+(50-100)*5\right]\right)$

$\theta_1 = 10 - 0.01\left(\frac{2}{5}(-550)\right)$

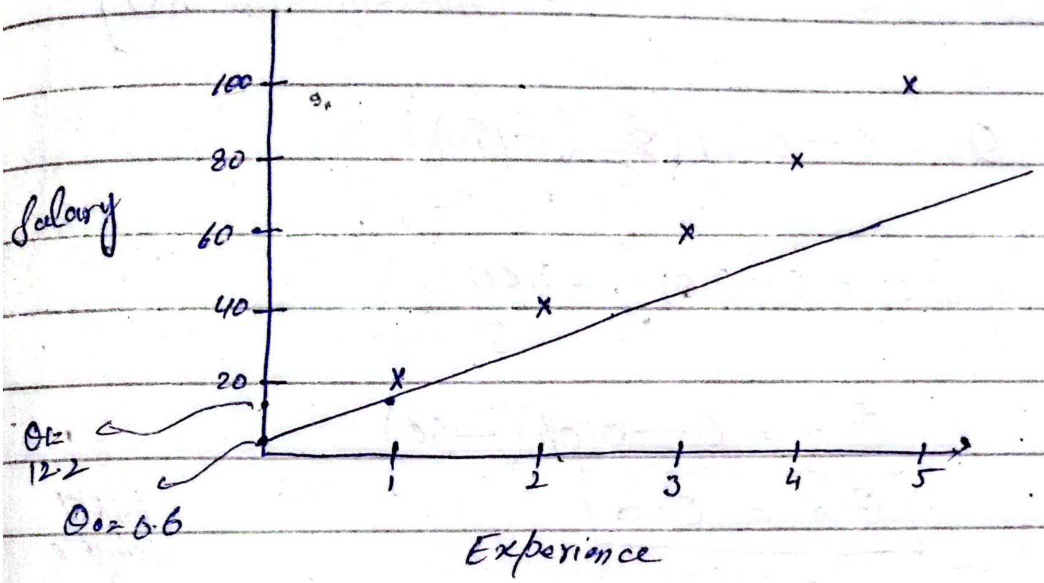
$\theta_1 = 10 - 0.01\left(-\frac{1100}{5}\right)$

$\theta_1 = 10 - 0.01(-220)$

$\boxed{\theta_1 = 12.2}$

Step 6 : Draw Graph with update θ_0 , θ_1 Value

$\theta_0 = 0.6$, $\theta_1 = 12.2$



Repeat Step 3, 4, 5, Until Cost / Error is minimize

Imp: اگر error ہمارے پاس زیادہ آ رہا ہے تو اس کا مطلب ہے ہم α کی ویلیو کو decrease کریں گے
(کیونکہ ہمارا model بڑے بڑے steps لے رہا ہے)

Multiple linear Regression

⇐ اس میں ہم نے steps بالکل Same ہوتے ہیں لیکن اس میں Steps میں مختلف hypothesis ہوتا ہے۔ اور اس میں Linear Regression والے ہی ہوتے ہیں۔ اس میں ایک سے زیادہ features ہوتے ہیں

Step 1:

$$\hat{Y} = h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 \ldots \theta_n x_n$$

$$h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 \ldots \theta_n x_n$$

Predict

⇐ اس کو Multiple linear Regression اس لیے بولتے ہیں کیونکہ اس میں ہمارے پاس multiple Independent Variable ہوتے ہیں۔

x_1 Size	x_2 Rooms	x_3 house Story	actual Price	$\hat{Y}$ predict
12	2	2	150 \$	
5	3	1	200 \$	
7	4	3	300 \$	

Step 2:

⇐ اس میں Cost function وہی use ہوتا ہے۔

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (\hat{Y}^i - Y^i)^2$$

predict actual

⇐ اس میں $\theta_0, \theta_1, \theta_2 \ldots \theta_n$ کو Gradient Descent کی ویلیو جیسے پہلے use کیا ہے ویسے ہی ہوگا۔

Step 3:

$$\theta_0 = \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1, \theta_2 \ldots \theta_n)$$

$$\theta_0 = \theta_0 - \alpha \left(\frac{2}{2m} \sum_{i=1}^{m} (\hat{Y} - Y) x_0^i\right)$$ always $x_0 = 1$

$$\theta_1 = \theta_1 - \alpha \left(\frac{2}{2m} \sum_{i=1}^{m} (\hat{Y} - Y) x_1^i\right)$$ Input feature 1

$$\theta_2 = \theta_2 - \alpha \left(\frac{2}{2m} \sum_{i=1}^{m} (\hat{Y} - Y) x_2^i\right)$$ instance, Input feature 2, column no

⋮

Lecture 11 Polynomial Regression (Slides)

جب ہمارے پاس data ہو جو higher degree کا ہو تو ہم جاتے ہیں
یہی وجہ ہے کہ ہم اس regression کو degree term کی Polynomial
بولتے ہیں

Polynomial کو w.r.t thetas کے ہم linear Regression ہی بولیں گے۔
کیونکہ thetas کی پاور one ہی ہے۔
Polynomial اس لیے کہتے ہیں کیونکہ Independent Variable کی degree بڑھائی گئی ہے۔
ہم اس طرح 2 سے زیادہ بھی کر سکتے ہیں جو curve fit کرتے ہیں data کو
نوٹ: اس کے لیے data کو 2D میں visualize کر کے دیکھ لیتے ہیں

Step 1: Find hypothesis and assign randomly theta

$\Rightarrow h_\theta(x) = \theta_0 + \theta_1 x_1$ linear

$\Rightarrow h_\theta(x) = \theta_0 + \theta_1 x_0 + \theta_2 x^2$ Quadratic

$\Rightarrow h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$ Cubic

$\Rightarrow h_\theta(x) = \theta_0 + \theta_1 \sqrt{x}$ Square root term

Step 2: Calculate Cost function

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (\hat{Y} - Y)^2$$

Step 3: Gradient Descent Algorithm

$$\theta_0 = \theta_0 - \alpha \frac{d}{d}(\text{Cost function})$$

Regularization (use for overfitting)

handle

اس میں ہم features کو reduce نہیں کرتے بلکہ ان feature
کی importance کو control کرتے ہیں thetas کی value
کو control کرتے ہیں۔

$$(\theta_1^2+\theta_2^2+\theta_3^2+\dots+\theta_n^2)$$

$$\text{Cost function } J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(\hat{Y}-Y)^2 + \lambda\sum_{j=1}^{m}\theta_j^2$$

$$2\lambda\theta_j$$

$$\text{Cost derivative } \frac{\partial}{\partial\theta_1} = \frac{1}{m}\sum_{i=1}^{m}(\hat{Y}-Y)x_j^i + \lambda\sum_{j=1}^{m}2\theta_j$$

$$\theta_{1_{new}} = \theta_{1\,old} - \alpha\left(J(\theta)\right)$$

For example:

$$h\theta(x) = \theta_0 + \theta_1 x$$

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(\hat{Y}-Y)^2 + 100\,\theta_1^2$$

$$\frac{d}{d\theta_1} = \frac{1}{m}\sum_{i=1}^{m}(\hat{Y}-Y) + 200\,\theta_1$$

$$\theta_1 = \theta_{1\,old} - \alpha\left(J(\theta)\right)$$

⇐ اس میں ہم λ کو ہم regularization پیرامیٹر بھی بولتے ہیں۔
Cost function کو ہم objective function بھی بولتے ہیں۔
⇐ اگر ہم λ کی ویلیو بڑی رکھیں گے تو ہم underfitting کی طرف
move ہونا شروع ہو جائیں گے۔
⇐ اگر λ کی ویلیو θ چھوٹی ہوگی تو ہمارا model میں سے
overfitting کم نہیں ہوگی۔

Reglurization ⇒

$$\theta_{j\,new} = \theta_{j\,old}\left(1-\alpha\frac{\lambda}{m}\right) - \alpha\left(\frac{1}{m}\sum_{i=1}^{m}(\hat{Y}-Y)x_j^i\right)$$

Ridge & Lasso Regression

1) Ridge Regression (L_2 Regularization)

Logist

$$\text{Cost} = -\frac{1}{m}\sum_{i=1}^{m} Y \log(\hat{Y}) + (1-Y)\log(1-\hat{Y}) + \lambda \sum_{j=1}^{n} \theta_j^2$$

Linear

$$\text{Cost} = \frac{1}{2m}\sum_{i=1}^{m} (\hat{Y}-Y)^2 x_i^j + \lambda \sum_{j=1}^{n} \theta_j^2$$

2) Lasso Regression (L_1 Regularzation)

Logist

$$\text{Cost} = -\frac{1}{m}\sum_{i=1}^{m} \hat{Y}\log(\hat{Y}) + (1-Y)\log(1-\hat{Y}) + \lambda \sum_{j=1}^{n} |\theta_j|$$

Linear

$$\text{Cost} = \frac{1}{2m}\sum_{i=1}^{m} (\hat{Y}-Y)^2 x_i^j + \lambda \sum_{j=1}^{n} |\theta_j|$$

Ridge and lasso Regression used:

1) Preventing the overfitting
2) Help to feature selection