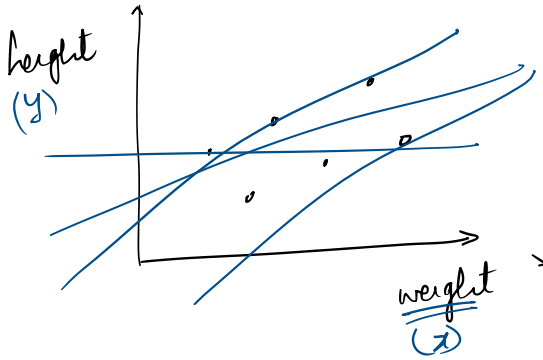
Gradient Descent →



parameters

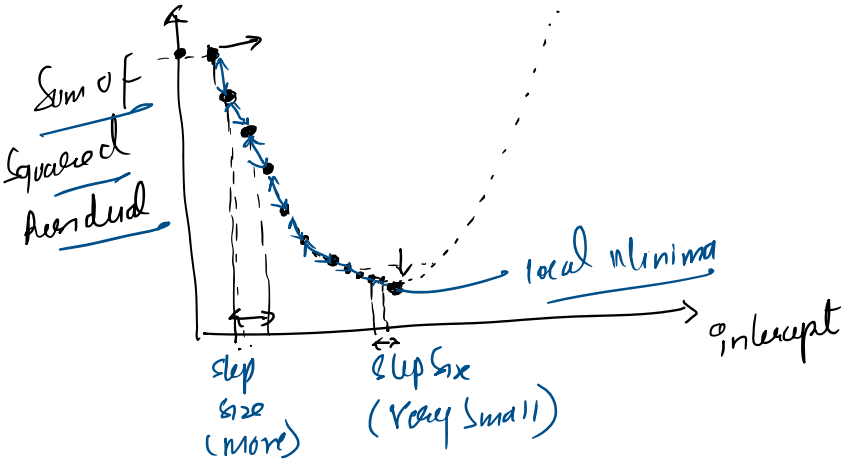
predicted height = intercept + slope * weight

$y = c + mx$

* Here we can have many line with different intercept and slope values

* We want a line with intercept and slope such that the Cost function should give Minimum value.
* Let the cost function be Sum of Squared Error (Residual Error)

* Let us Consider only intercept
* Lets plot a graph of Intercept V/s Sum of Squared Residual Error.



To find Minimum value for Sum of Squared Residual Error, we will start with very small value of intercept and will increase the value by very small amount.

Ex

x	y
0.5	1.4
2.3	1.9

Let the cost = Sum of Squared Error Fn be (Residual).

Ex

0.5	1.4
2.3	1.9
2.9	3.2

↑ ↑ actual

Let the cost = Sum of squared ... Fn be (Residual).

$$\text{Sum of Squared Residual Error} = (1.4 - (\text{intercept} + \text{slope} * 0.5))^2 + (1.9 - (\text{intercept} + \text{slope} * 2.3))^2 + (3.2 - (\text{intercept} + \text{slope} * 2.9))^2$$

Here we want to find value of intercept and slope that give minimum Sum of squared Error.

Step 1 → Find $\frac{d}{d\,\text{intercept}}$ (Sum of Squared Residual Error)

$\frac{d}{d\,\text{slope}}$ (Sum of Squared Residual Error).

$$\text{Sum of Squared Residual Error (SSE)} = (1.4 - (\text{intercept} + \text{slope} * 0.5))^2 + (1.9 - (\text{intercept} + \text{slope} * 2.3))^2 + (3.2 - (\text{intercept} + \text{slope} * 2.9))^2$$

Using chain Rule $\left[\frac{d}{dx} x^2 = 2x \frac{d}{dx} x\right]$

$$\frac{d(\text{Sum of Squared Error})}{d\,\text{intercept}} = 2 * (1.4 - (\text{intercept} + \text{slope} * 0.5)) * \frac{d}{d\,\text{intercept}} (1.4 - (\text{intercept} + \text{slope} * 0.5))$$

$$\frac{d}{d\,\text{intercept}} (1.4 - (\text{intercept} + \text{slope} * 0.5)) \quad (-1)$$

$$+ 2 * (1.9 - (\text{intercept} + \text{slope} * 2.3)) \frac{d}{d\,\text{intercept}} (1.9 - (\text{intercept} + \text{slope} * 2.3)) \quad (-1)$$

$$+ 2 * (3.2 - (\text{intercept} + \text{slope} * 2.9)) \frac{d}{d\,\text{intercept}} (3.2 - (\text{intercept} + \text{slope} * 2.9)) \quad (-1)$$

$$\frac{d}{d\,intercept}\left(3.2 - (intercept + slope * 2.9)\right)$$

$$(-1)$$

$$\frac{d(SSE)}{d\,intercept} = (-2)(1.4 - (intercept + slope * 0.5)) + (-2)(1.9 - (intercept + slope * 2.3)) + (-2)(3.2 - (intercept + slope * 2.9)).$$

Eq 1.

Sum of Squared Residual Error (SSE)

$$= (1.4 - (intercept + slope * 0.5))^2 + (1.9 - (intercept + slope * 2.3))^2 + (3.2 - (intercept + slope * 2.9))^2$$

Now

$$\frac{d(SSE)}{d\,slope} = \frac{d}{d\,slope}(1.4 - (intercept + slope * 0.5))^2 + \frac{d}{d\,slope}(1.9 - (intercept + slope * 2.3))^2 + \frac{d}{d\,slope}(3.2 - (intercept + slope * 2.9))^2.$$

$$\frac{d(SSE)}{d\,slope} = (-2)(0.5)(1.4 - (intercept + slope * 0.5)) + (-2)(2.3)(1.9 - (intercept + slope * 2.3)) + (-2)(2.9)(3.2 - (intercept + slope * 2.9)).$$

Eq 2.

We got the above 2 equations.

Let start with Random values of intercept & slope

Let intercept = 0, slope = 1

Put intercept = 0 & slope = 1 in Eq 1 & Eq 2

Substitute $\varphi_{intercept} = 0$ & slope = 1 in Eq 1 & Eq 2

$\frac{d(SSE)}{d\,slope} = -0.8$ ⇒ slope of line wrt parameter slope

$\frac{d(SSE)}{d\,intercept} = -1.6$ ⇒ slope of line wrt parameter $\varphi_{intercept}$.

For step size

New step size = slope of line at point * learning Rate

* Learning Rate ⇒ takes smaller value [10^{-6} to 1.0]

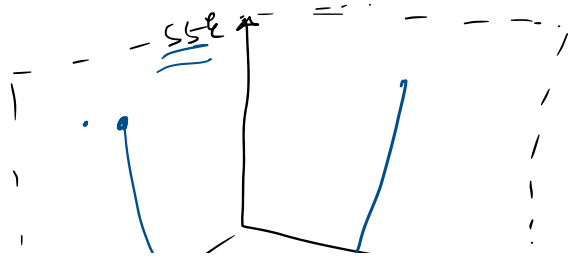
Let us take learning Rate = 0.01

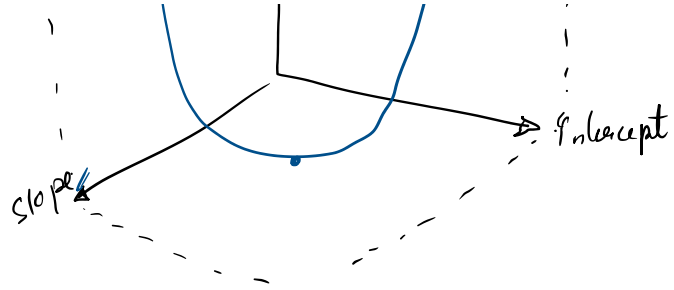
$\text{stepSize}_{(\varphi_{intercept})} = \frac{d(SSE)}{d\,intercept} * 0.01 = -1.6 * 0.01 = -0.016$

$\text{stepSize}_{(slope)} = \frac{d(SSE)}{d\,slope} * 0.01 = -0.8 * 0.01 = -0.008$

New Intercept = old $\varphi_{intercept}$ − $\text{stepSize}_{(intercept)}$ = 0 − (−0.016) = 0.016

New slope = old slope − $\text{stepSize}_{(slope)}$ = 1 − (−0.008) = 1.008





[Substitute the new intercept and slope in $\frac{d(SSE)}{d\,intercept}$ & $\frac{d(SSE)}{d\,slope}$ and Calculate new step size for intercept & slope
And again Calculate New Intercept & New slope]

* We will repeat until the step size becomes very small
  ⇒ 0.001

or Some maximum Number of steps is reached Ex: 1000

[For above Dataset Best fitting line will have
intercept = 0.95 & slope = 0.64]

[This is How Gradient Descent Optimizes Parameters]

Summary -> Gradient Descent

1. Get Eq^n of line [Identify Simple Linear or Polynomial]
2. Want intercept & slope value that Minimizes Cost f^n
3. Identify Cost (loss) function. [we took SSE]
4. Find Eq for $\frac{d(SSE)}{d\,intercept}$ & $\frac{d(SSE)}{d\,slope}$
5. Start with Some Initial value of slope & intercept.
6. Find the value of $\frac{d(SSE)}{d\,intercept}$ & $\frac{d(SSE)}{d\,slope}$

(6) Find the value of $\frac{d(SSE)}{d\,intercept} * \frac{d(SSE)}{d\,slope}$

(7) Calculate $\text{StepSize}_{(intercept)} = \frac{d(SSE)}{d\,intercept} * \text{learning rate}$

$\text{StepSize}_{(slope)} = \frac{d(SSE)}{d\,slope} * \text{learning rate}$.

(8) Update intercept = old intercept − $\text{StepSize}_{(intercept)}$

Update slope = old slope − StepSize (slope)

(9) Repeat step 6 − 8 untill stepSize is very small

or Number of iterations $\leq$ 1000

Gradient Boost for Regression

* Gradient Boost start with Single Leaf instead of Tree/stump.

* Leaf represents initial guess for the weights of the Samples.

Consider

Height	Favorite color	Gender	Weight (Actual)
1.6	Blue	male	88
1.6	Green	Female	76
1.5	Blue	Female	56
1.8	Red	male	73
1.5	Green	male	77
1.4	Blue	Female	57

Step1 => First let make Initial Guess = Weight (Average) = 71.2

Step2 => Like Adaboost, Even Gradient Boost builds tree based on previous Tree

But unlike Adaboost, the Gradient Boost tree are larger than Stump

[However the Height is still Restricted [No of Leaf 8 → 32 in Each Tree]

* Gradient Boost will build Another Tree based on Error of previous Tree.

* [Gradient Boost continous to build Tree in this fashion untill it has made number of Trees you asked for or the Additional tree fails to improve the fit]

⇒ Step 3 : Initial Guess = weight (Average) = 71.2

Predicted value = 71.2 + Tree1 + Tree2 ...

71.2 ↑ Initial guess

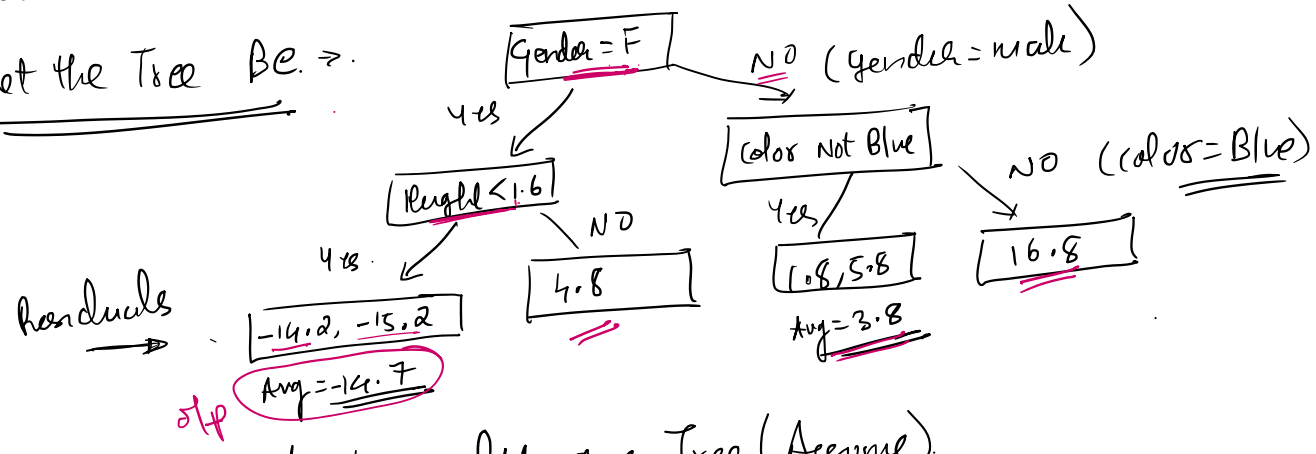
Tree1 + Tree2 ... → Boosting

Height	Favorite color	Gender	Weight	Residual (Observed - Average)
1.6	Blue	male	88	16.8
1.6	Green	Female	76	4.8
1.5	Blue	Female	56	-15.2
1.8	Red	male	73	1.8
1.5	Green	male	77	5.8
1.4	Blue	Female	57	-14.2

Initial 71.2

Let Build a Tree using Residual and not weight

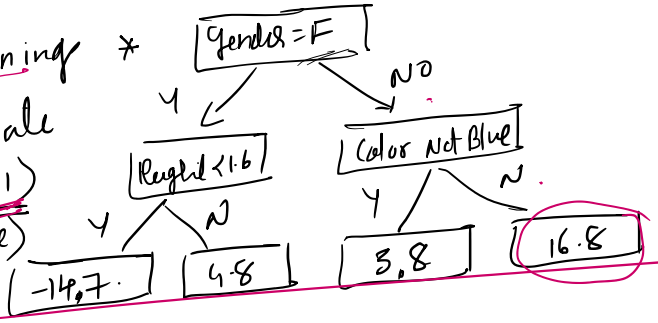
Let the Tree Be. =>



If there will be only one Tree (Assume)

Then.

Predicted value = Initial Guess + Learning Rate (0.1) (Assume) * Gender = F



Now first Sample = | 1.6 | Blue | male | ? weight |

Updated = 71.2 + 0.1 * 16.8 = 72.88

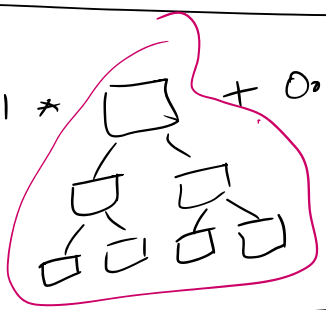
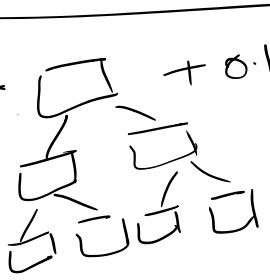
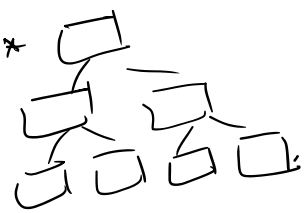
Predicted value = $71.2 + 0.1 * 16.8 = 72.88$

* Repeat this for every sample & calculate Predicted value for each sample
  → call it as New Prediction.

Height	Favorite color	Gender	Weight (Actual)	Residual	New Prediction	New Residual
1.6	Blue	male	88	16.8 ✓	72.88	R_1 ✓
1.6	Green	Female	76	-4.8	P_1 ✓	R_2 ✓
1.5	Blue	Female	56	-15.2	P_2 ✓	R_3 ✓
1.8	Red	male	73	1.8	P_3 ✓	R_4 ✓
1.5	Green	male	77	5.8	P_4 ✓	R_5 ✓
1.4	Blue	Female	57	-14.2	P_5 ✓	R_6 ✓
					71.2	

* Now find New Residual for New Prediction for all the Samples
* Using this New Residual Construct New Tree and so on Untill the New Tree does not make Significant Contribution to the New Prediction.

* Let us say we have 3 Tree for above Dataset →

Final Prediction = Initial value + 0.1 *  + 0.1 *  + 0.1 * 

[Gradient Boost Principle → Taking lot of Small steps in Right Direction will result into better Predictions]