# Ordinary Least Squares (OLS) & Gradient Descent

## Linear Regression (OLS: Ordinary Least Square)

| Symbols | Meaning |
|---|---|
| $x$ | Independent variable data from observation |
| $\overline{x}$ | Mean of $x$ |
| $y$ | Dependent variable data from observation |
| $\overline{y}$ | Mean of $y$ |
| $\hat{y}$ | Estimate of $y$ by the regression model |
| $n$ | Number of observations |

**Steps:**

1. Get the difference (error): $(y\text{-}\hat{y})$
2. Square the difference: $(y\text{-}\hat{y})^2$
3. Take the sum for all data: $\sum (y-\hat{y})^2$

This is total error. Our objective is to keep this as minimum as possible.

## Linear Regression (OLS: Ordinary Least Square)

$$Y = f(x) = 4(x-3)^2+5$$

$$SSE = f(?) = \sum (y-\hat{y})^2 = \sum (y - mx - c)^2$$

$$SSE = f(?) = \sum (y-\hat{y})^2 = \sum (y - \theta_1 x - \theta_0)^2$$

$$SSE = f(?) = \sum (y-\hat{y})^2 = \sum (y - \beta_1 x - \beta_0)^2$$

$$SSE = f(?) = \sum (y-\hat{y})^2 = \sum (y - ax - b)^2$$

$$SSE = f(?) = \sum_i^n (y_i-\hat{y}_i)^2 = \sum_i^n (y_i - ax_i - b)^2$$

## Linear Regression (OLS: Ordinary Least Square)

$$Y = f(x) = 4(x-3)^2+5$$

$$SSE = f(m,c) = \sum (y-\hat{y})^2 = \sum (y - mx - c)^2$$

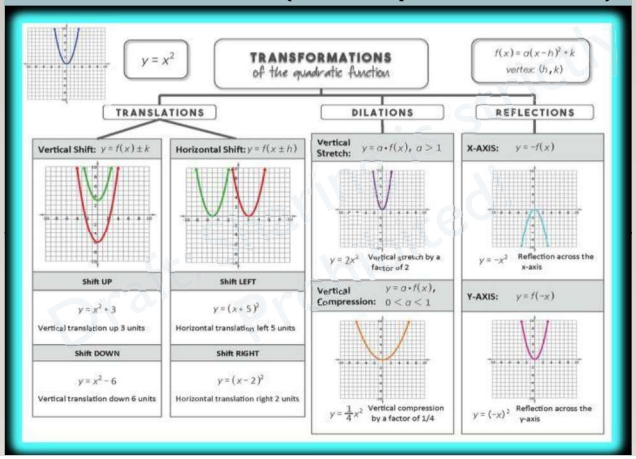
$$SSE = f(\theta_0,\theta_1) = \sum (y-\hat{y})^2 = \sum (y - \theta_1 x - \theta_0)^2$$

$$SSE = f(\beta_0,\beta_1) = \sum (y-\hat{y})^2 = \sum (y - \beta_1 x - \beta_0)^2$$

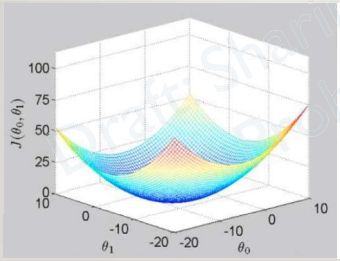
$$SSE = f(a,b) = \sum (y-\hat{y})^2 = \sum (y - ax - b)^2$$

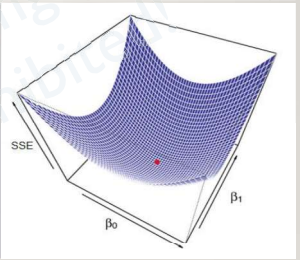
$$SSE = f(a,b) = \sum_i^n (y_i-\hat{y}_i)^2 = \sum_i^n (y_i - ax_i - b)^2$$

## Quadratic Functions (one independent variable)

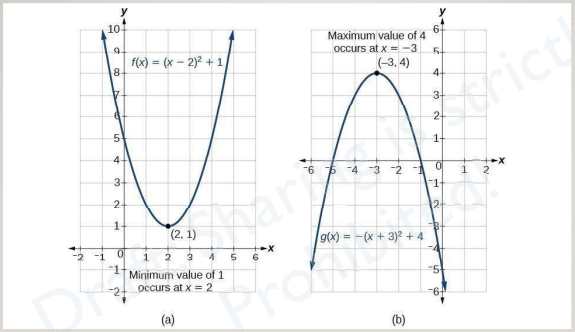


## Quadratic Functions (Two independent variables)

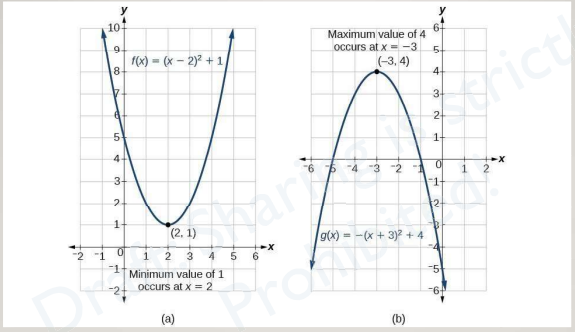




## Minimum value ofY



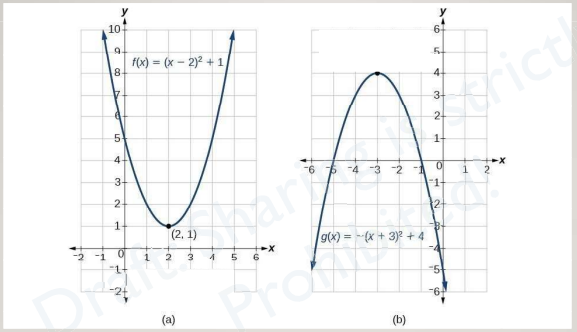

Give me $x$, where the value of $y$ is minimum.

## Minimum value ofY




Differentiate $y$, Set its value to 0, solve the equation to find the value of $x$.

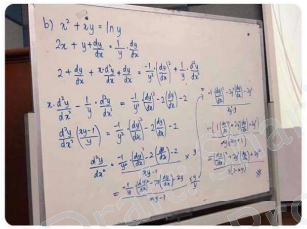
## Minimum value ofY




Give me $x$, where the value of $y$ is minimum.

## Minimum value ofY

I miss the brain that can understand this...



Thicc and Tired
@flygeriangirl_

Can't believe there was actually a time in my life when I could solve this. What was the reason?

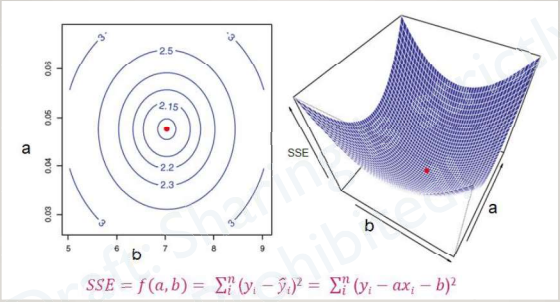
I dont understand why this kind of crap is mandatory but they dont teach us about taxs or how to rent/own your own home, or even how to reasonably budget your money.... Pretty sad honeestly

Like · Reply · 3h 10

Wait. You guys had a brain that could solve this?!

Like · Reply · 21h 502

## Minimum value of SSE




$$SSE = f(a,b) = \sum_i^n (y_i - \hat{y}_i)^2 = \sum_i^n (y_i - ax_i - b)^2$$

Give me $(a, b)$, where the value of SSE is minimum.

Differentiate SSE partially:

a) With respect to $a$, Set its value to 0, Solve the equation to find the value of $a$.
b) With respect to $b$, Set its value to 0, Solve the equation to find the value of $b$.

## Linear Regression (OLS: Ordinary Least Square)

Let us denote SSE as S for simplicity: $S = \sum (y-\hat{y})^2 = \sum (y-ax-b)^2$

$$\frac{\partial S}{\partial a} = 0 \qquad \frac{\partial S}{\partial b} = 0$$

$$\frac{\partial S}{\partial a} = \frac{\partial \left\{\sum (y-ax-b)^2\right\}}{\partial a} = 2\sum \{(y-ax-b)\cdot(0-x-0)\} \qquad \frac{\partial S}{\partial b} = \frac{\partial \left\{\sum (y-ax-b)^2\right\}}{\partial b} = 2\sum \{(y-ax-b)\cdot(0-0-1)\}$$

$$2\sum \{(y-ax-b)\cdot(-x)\} = 0 \qquad -2\sum \{(y-ax-b)\} = 0$$

$$\sum (-xy) + a\sum x^2 + b\sum x = 0 \qquad -\sum y + a\sum x + b\sum 1 = 0$$

$$\sum x = n\bar{x} \qquad \sum 1 = n \quad \sum x = n\bar{x} \quad \sum y = n\bar{y}$$

$$b = \frac{\sum xy - a\sum x^2}{n\bar{x}} \qquad -n\bar{y} + an\bar{x} + nb = 0 \quad a\bar{x} + b = \bar{y}$$



## Linear Regression (OLS: Ordinary Least Square)

Let us denote SSE as S for simplicity: $S = \sum (y-\hat{y})^2 = \sum (y-ax-b)^2$

$$\frac{\partial S}{\partial a} = 0 \qquad \frac{\partial S}{\partial b} = 0$$

$$\frac{\partial S}{\partial a} = \frac{\partial \left\{\sum (y-ax-b)^2\right\}}{\partial a} = 2\sum \{(y-ax-b)\cdot(0-x-0)\} \qquad \frac{\partial S}{\partial b} = \frac{\partial \left\{\sum (y-ax-b)^2\right\}}{\partial b} = 2\sum \{(y-ax-b)\cdot(0-0-1)\}$$

$$2\sum \{(y-ax-b)\cdot(-x)\} = 0 \qquad -2\sum \{(y-ax-b)\} = 0$$

$$\sum (-xy) + a\sum x^2 + b\sum x = 0 \qquad -\sum y + a\sum x + b\sum 1 = 0$$

$$\sum x = n\bar{x} \qquad \sum 1 = n \quad \sum x = n\bar{x} \quad \sum y = n\bar{y}$$

$$b = \frac{\sum xy - a\sum x^2}{n\bar{x}} \qquad -n\bar{y} + an\bar{x} + nb = 0 \quad a\bar{x} + b = \bar{y}$$

$$a\bar{x} + \frac{\sum xy}{n\bar{x}} - \frac{a\sum x^2}{n\bar{x}} = \bar{y}$$

$$a\left(\bar{x} - \frac{\sum x^2}{n\bar{x}}\right) + \frac{\sum xy}{n\bar{x}} = \bar{y}$$

$$a\left\{n\bar{x}^2 - \sum x^2\right\} + \sum xy = n\bar{x}\bar{y}$$

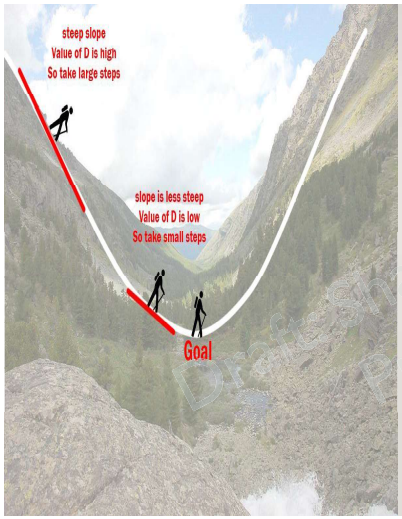
$$a = \frac{n\bar{x}\bar{y} - \sum xy}{\left(n\bar{x}^2 - \sum x^2\right)}$$

$$\hat{y} = slope * x + intercept$$

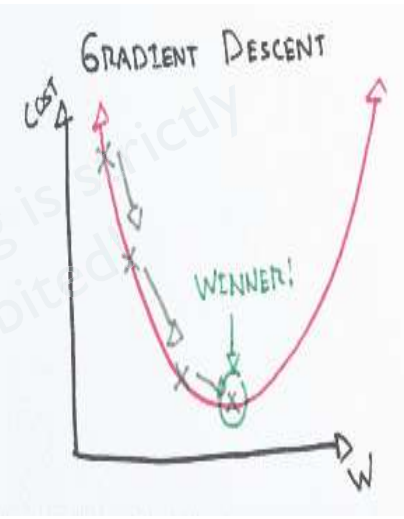
$$slope = \frac{n\sum xy - \sum x\sum y}{n\sum x^2 - (\sum x)^2}$$

$$intercept = \bar{y} - slope \cdot \bar{x}$$

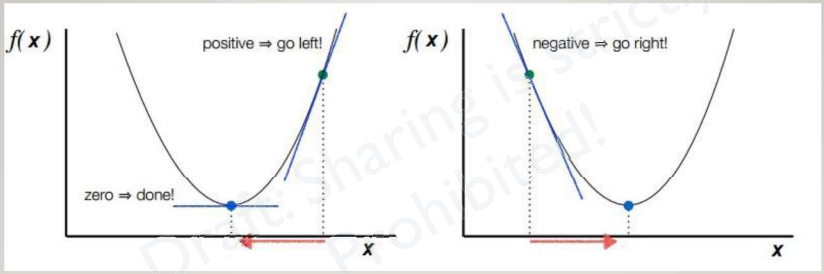
Mother of Dragons

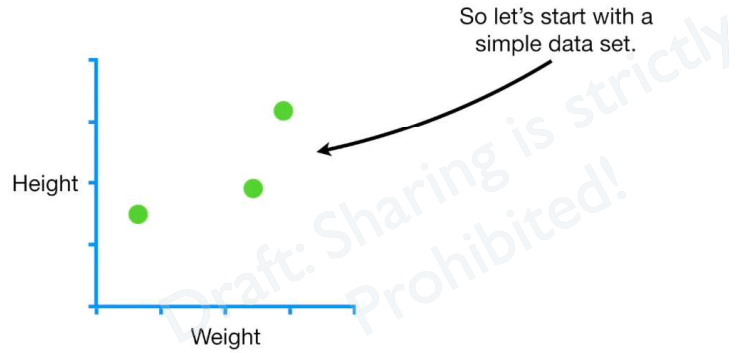


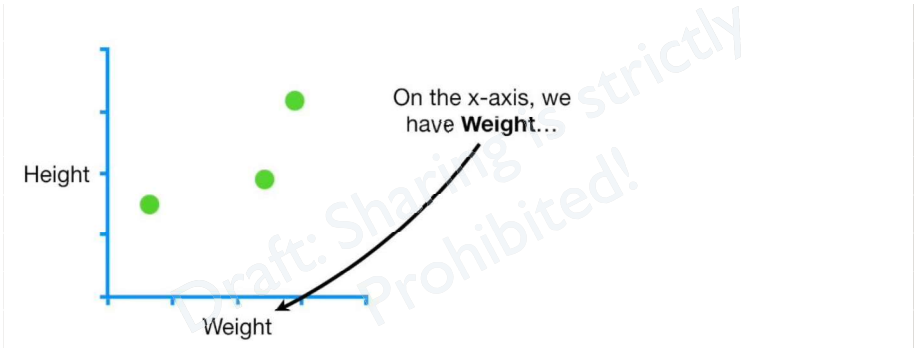

Mother of ML Algorithms

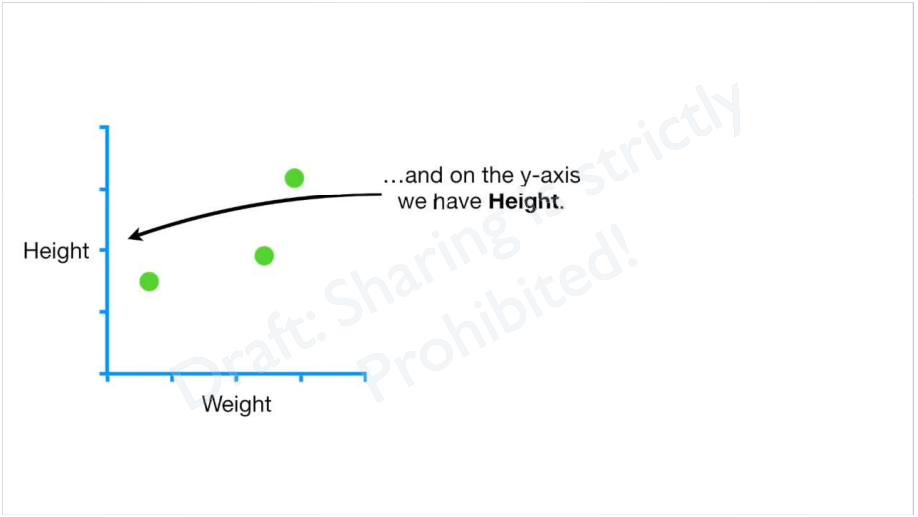


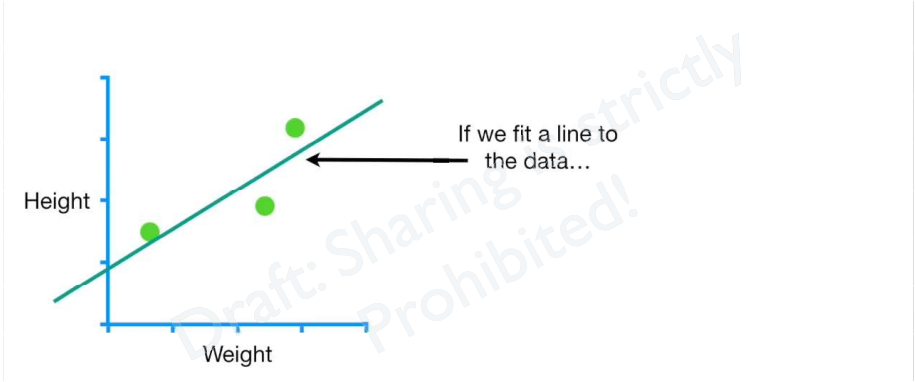

## Gradient Descent

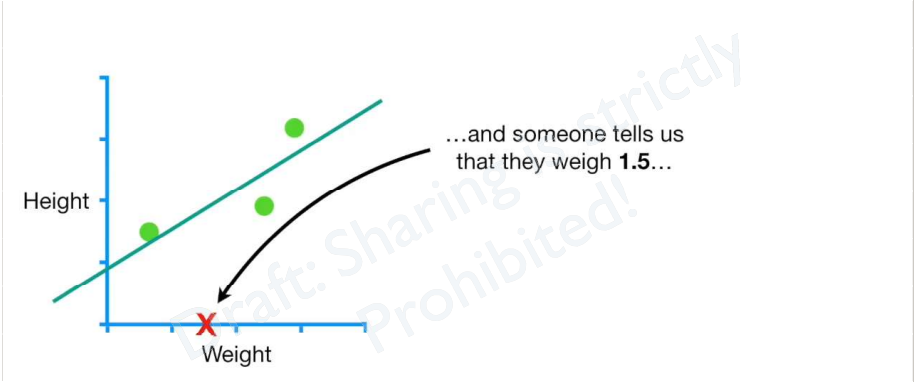


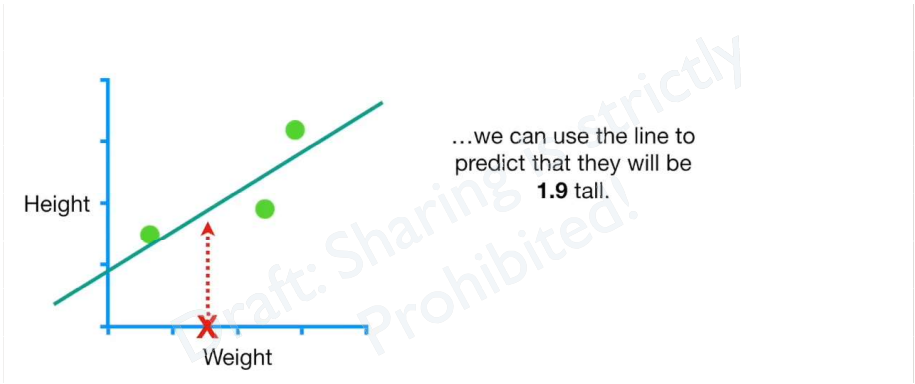



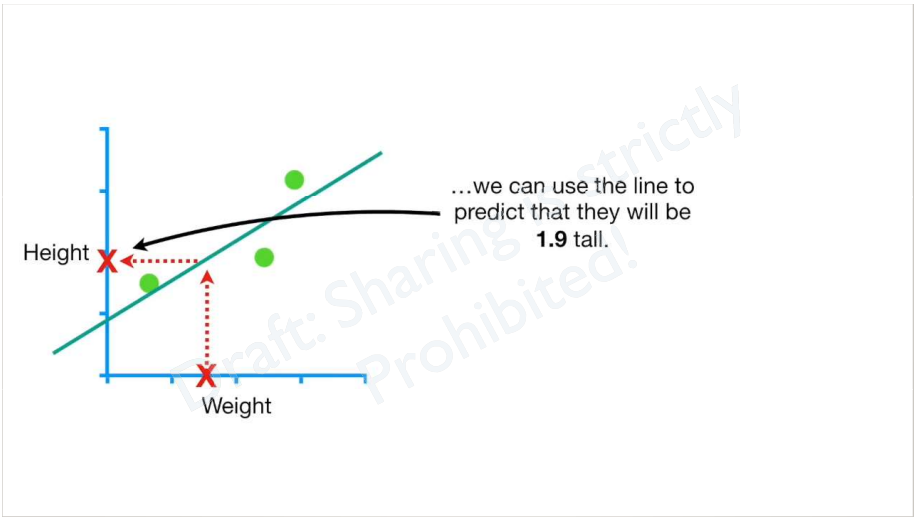


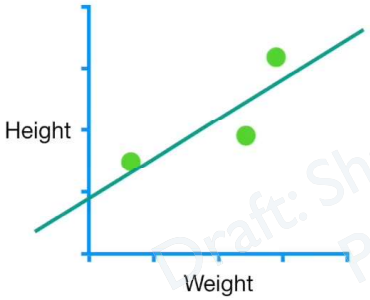



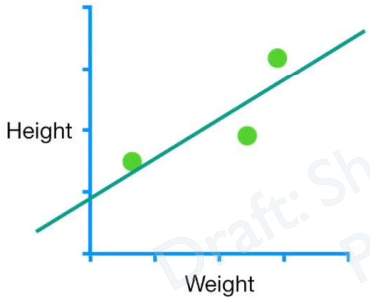
**Predicted Height** = intercept + slope × **Weight**




So let's learn how **Gradient Descent** can fit a line to data by finding the optimal values for the **Intercept** and the **Slope**.



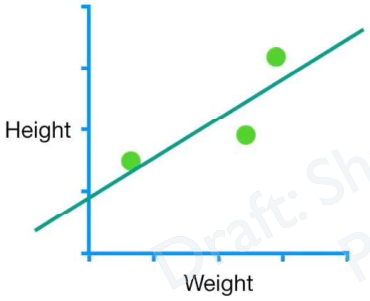
**Predicted Height** = intercept + slope × **Weight**




Actually, we'll start by using **Gradient Descent** to find the **Intercept**.



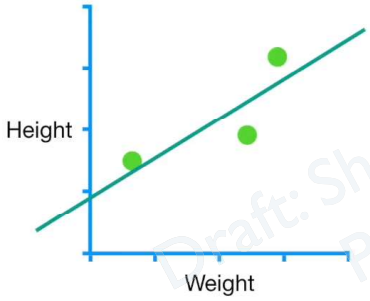
**Predicted Height** = intercept + slope × **Weight**




Then, once we understand how **Gradient Descent** works, we'll use it to solve for the **Intercept** *and* the **Slope**.



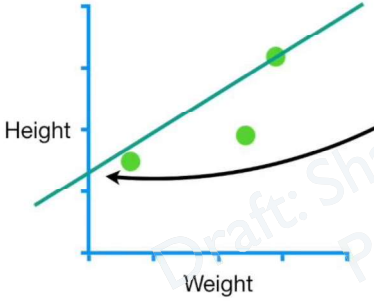
**Predicted Height** = intercept + slope × **Weight**




So for now, let's just plug in the **Least Squares** estimate for the **Slope**, **0.64**.



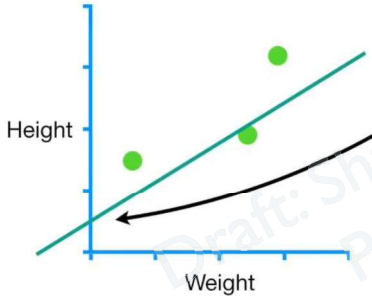
**Predicted Height** = intercept + 0.64 × **Weight**




...and we'll use **Gradient Descent** to find the the optimal value for the Intercept.

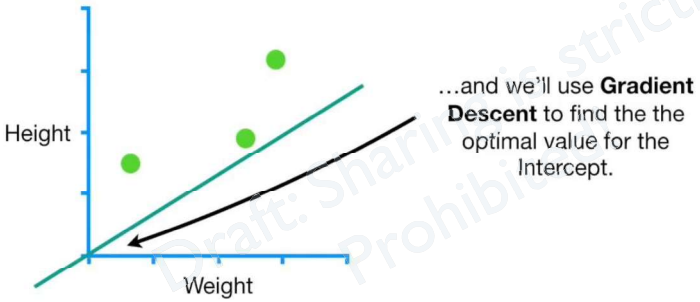


**Predicted Height** = intercept + 0.64 × **Weight**



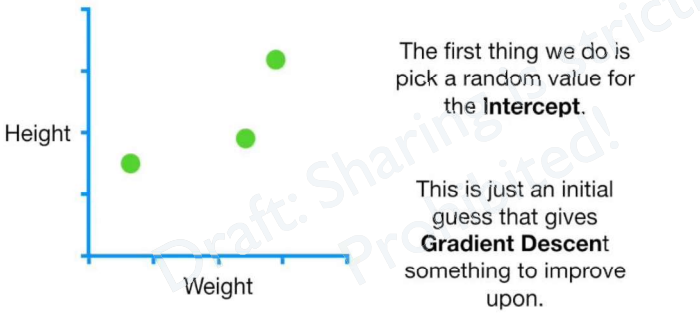

...and we'll use **Gradient Descent** to find the the optimal value for the Intercept.

**Predicted Height** = intercept + 0.64 × **Weight**



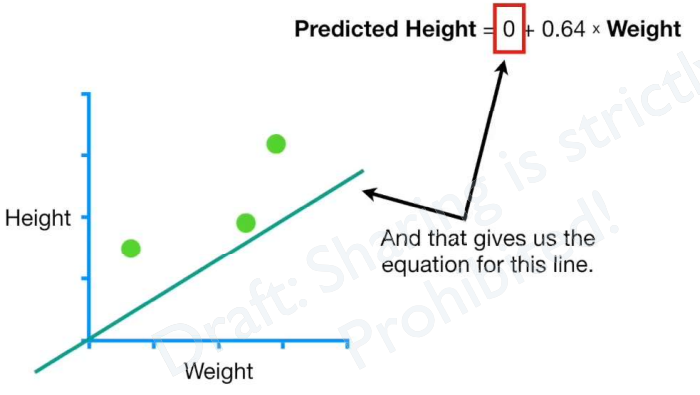



**Predicted Height** = intercept + 0.64 × **Weight**



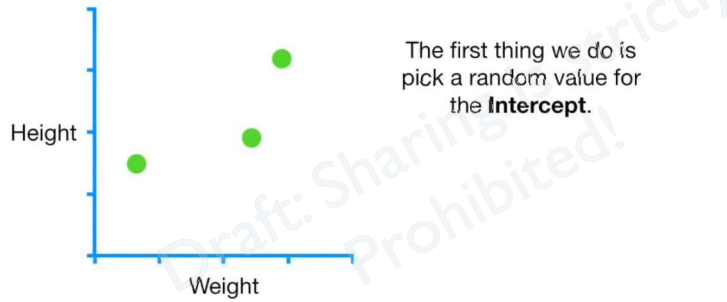



**Predicted Height** = 0 + 0.64 × **Weight**



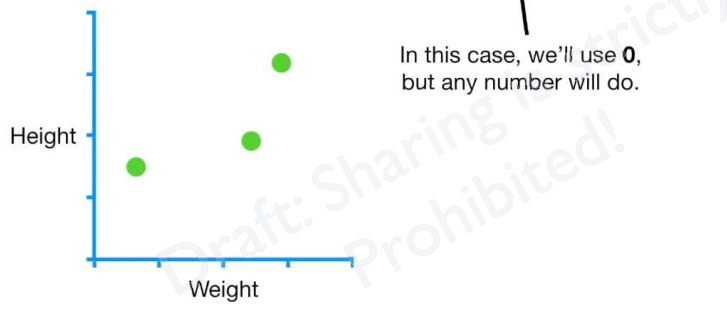



**Predicted Height** = intercept + 0.64 × **Weight**

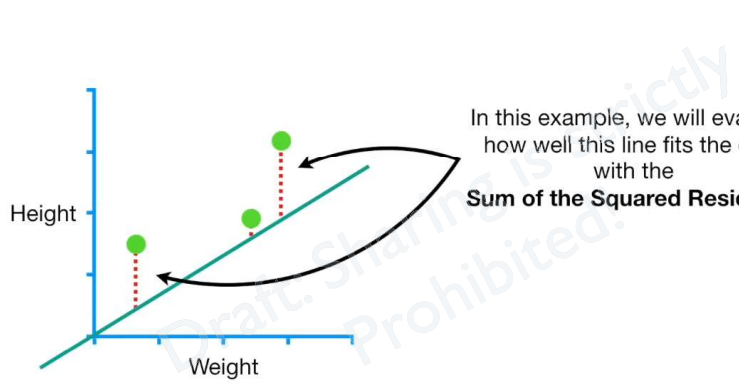


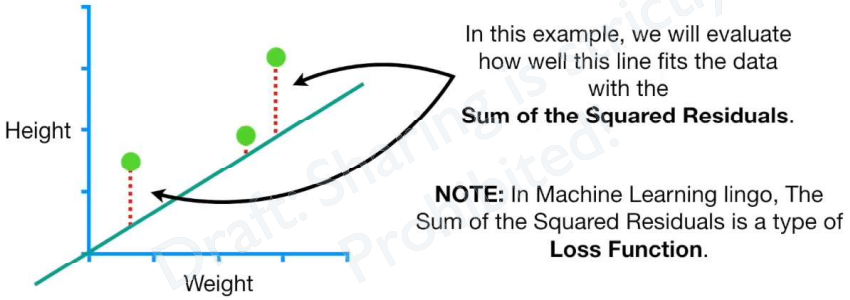


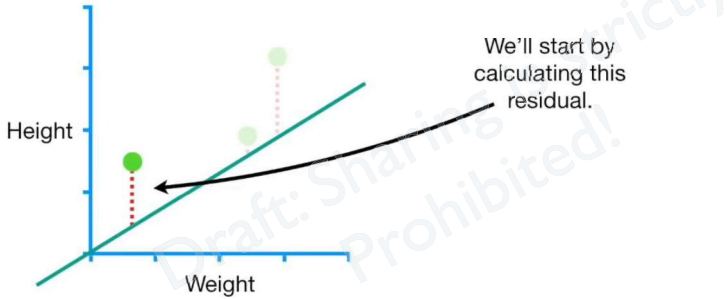

**Predicted Height** = 0 + 0.64 × **Weight**

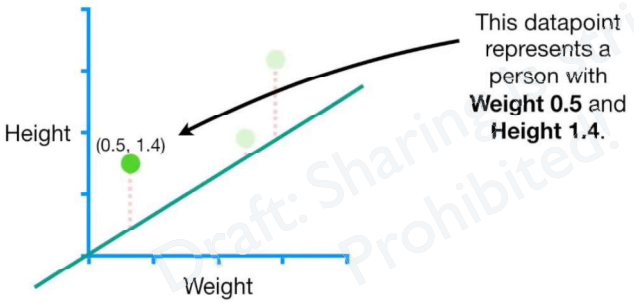


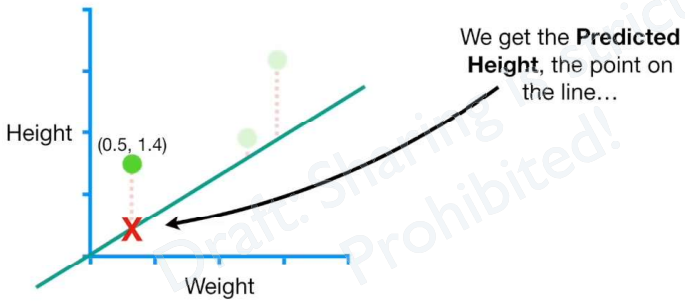


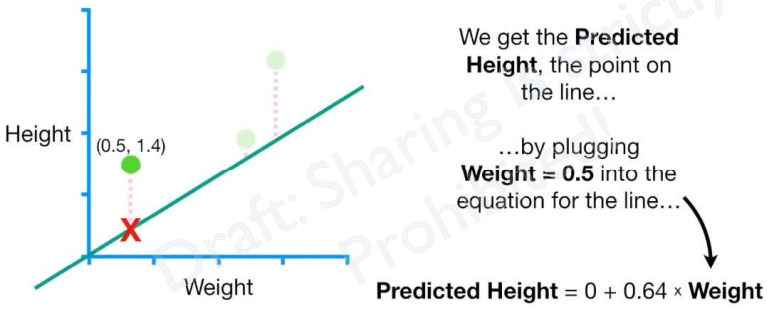



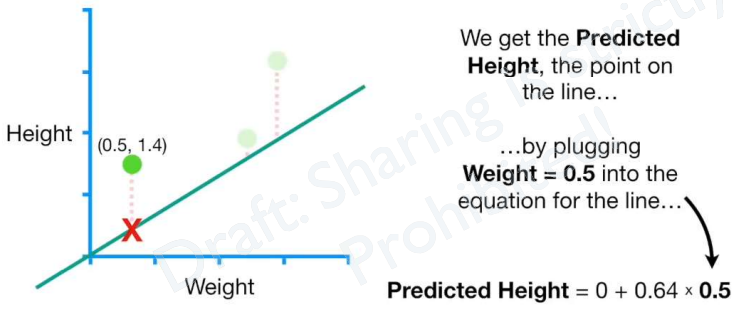


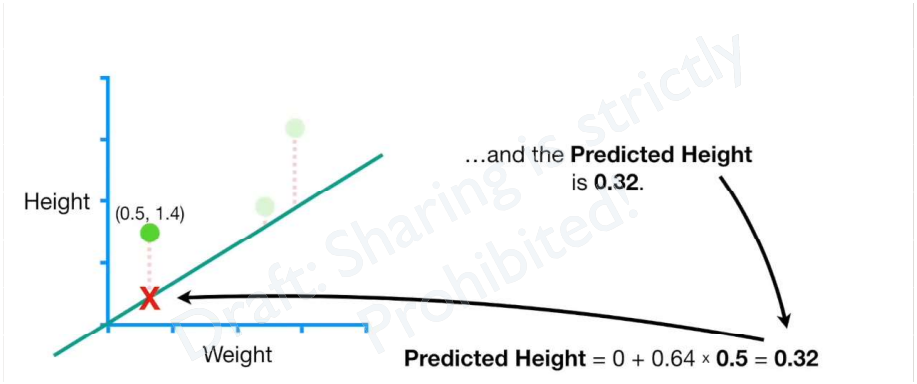



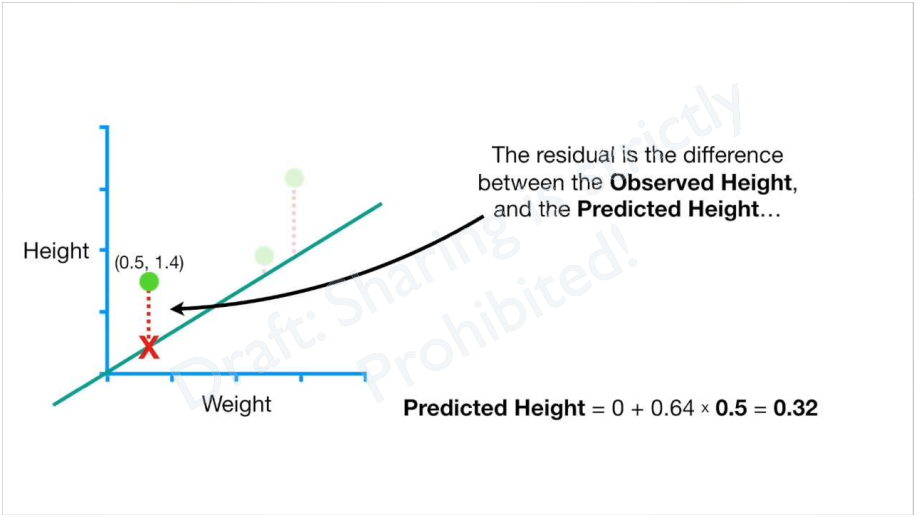



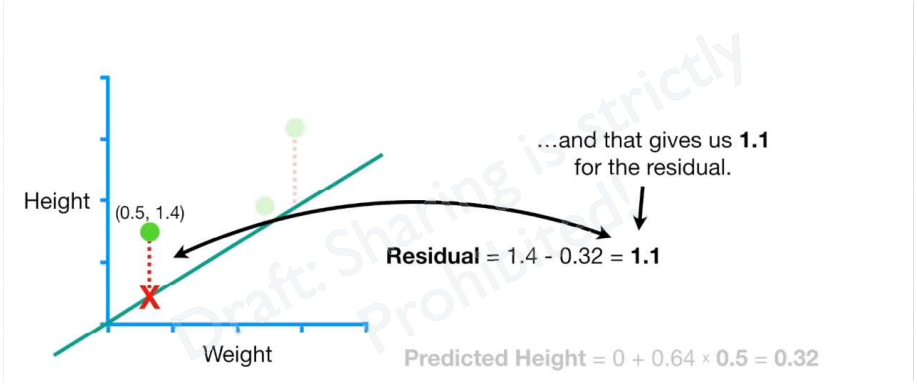



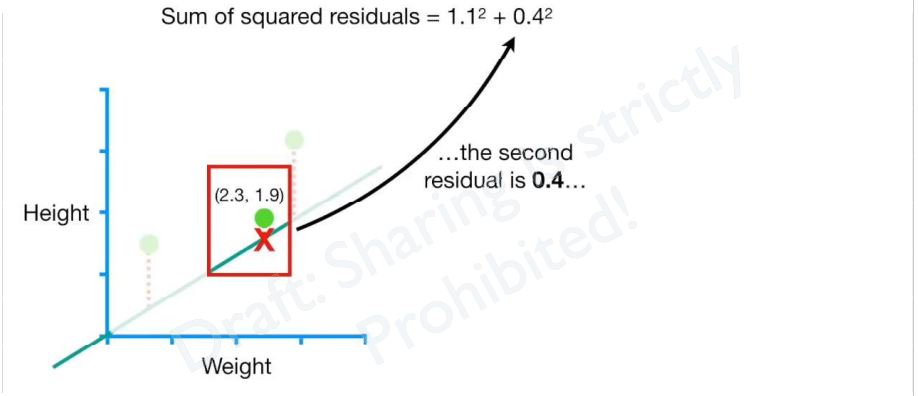



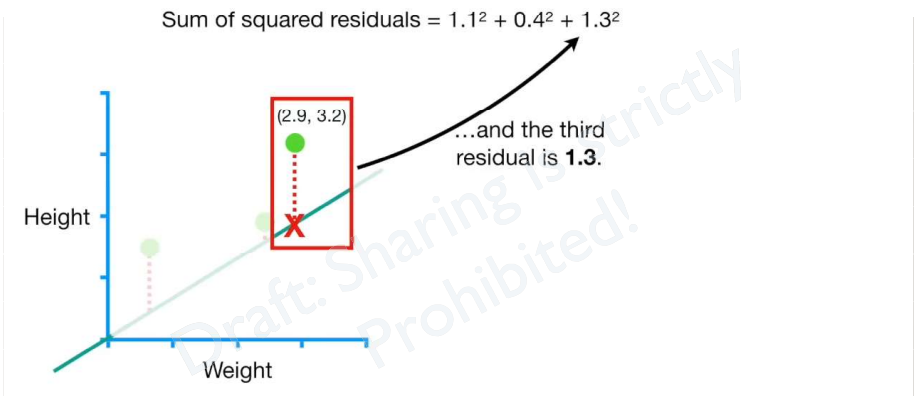



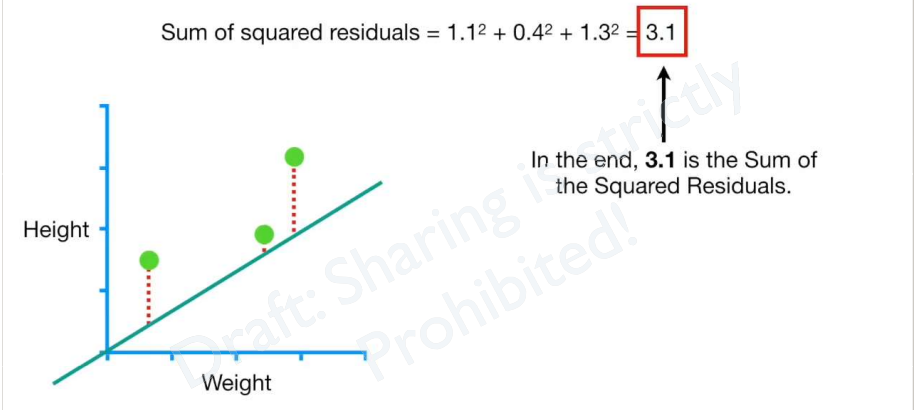


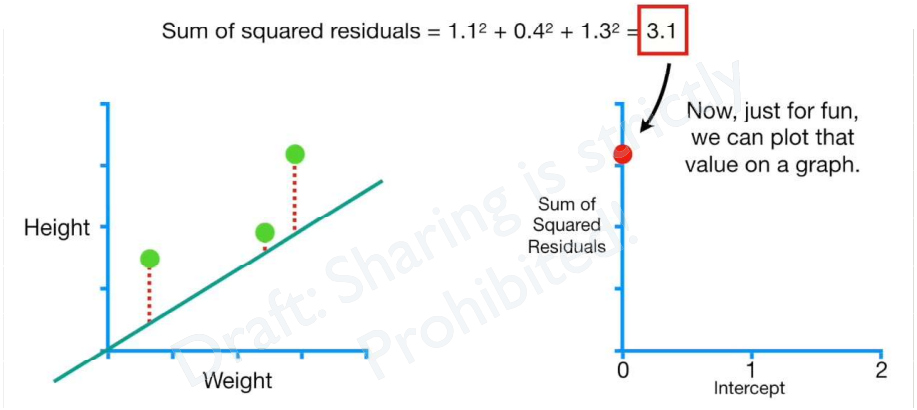



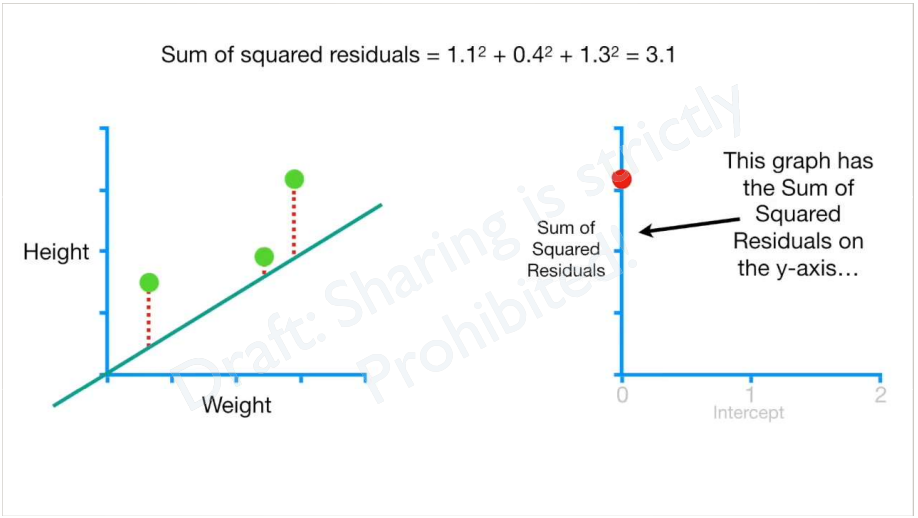



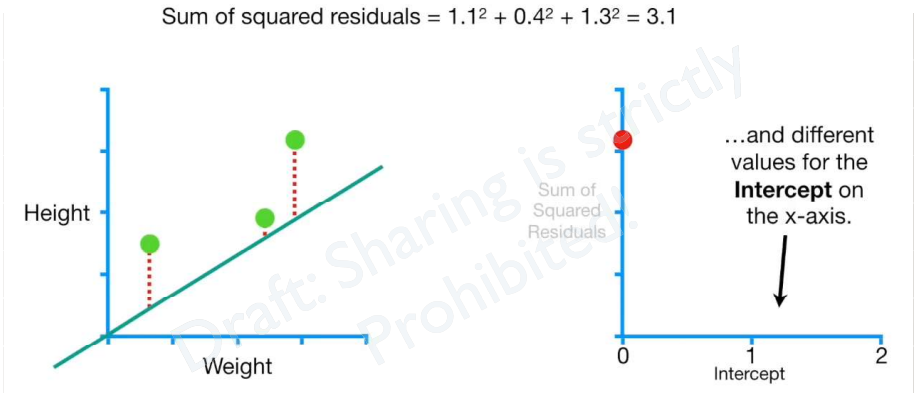



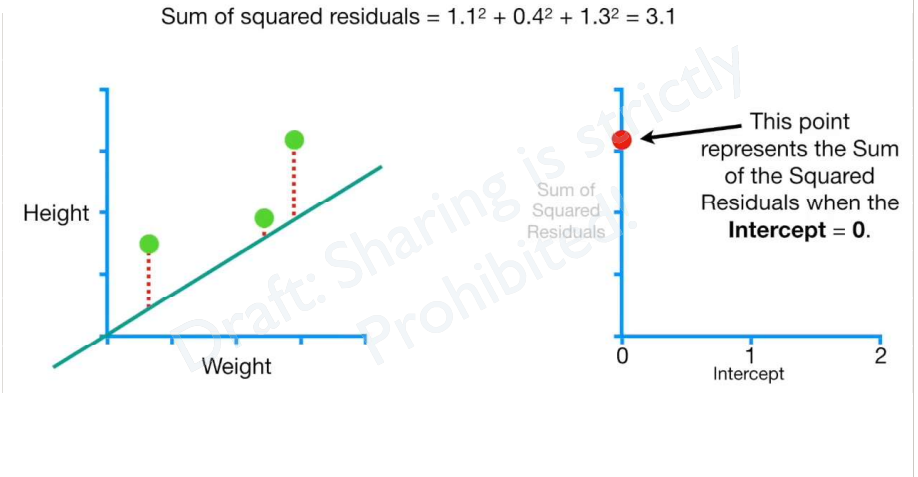



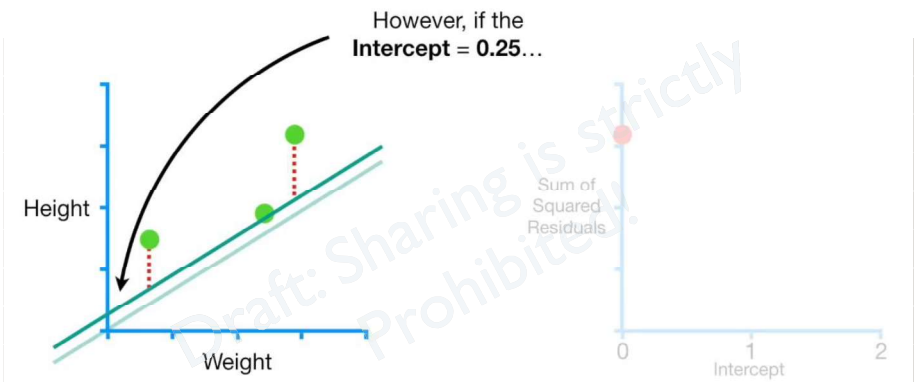



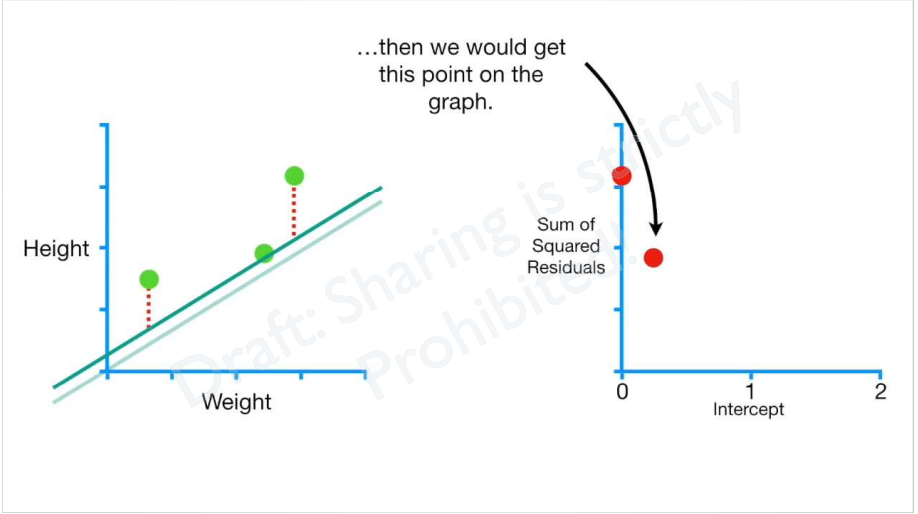


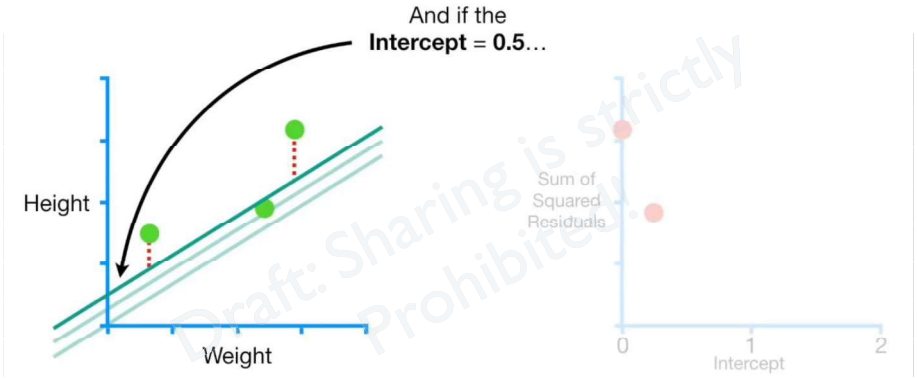



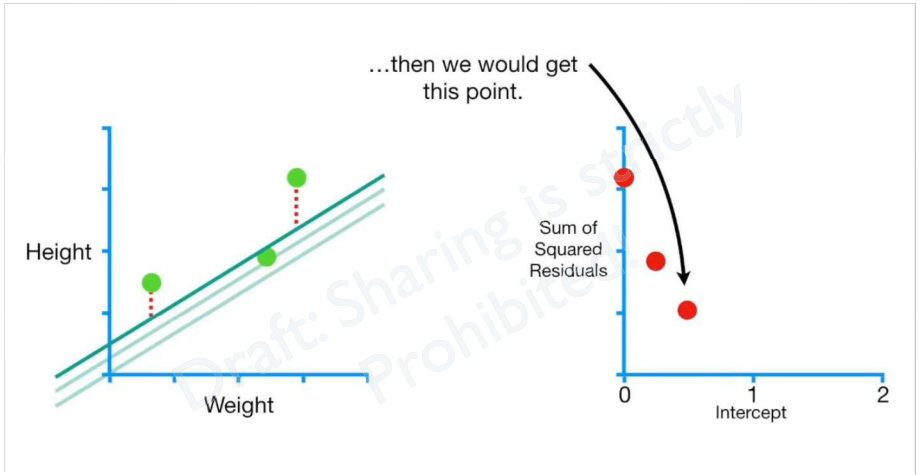



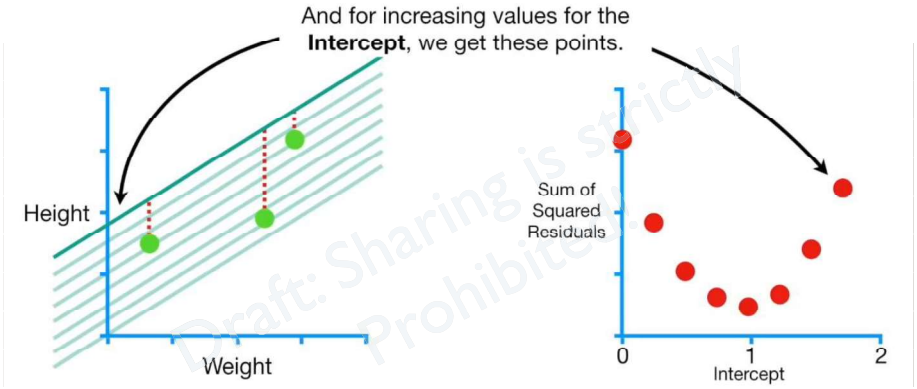



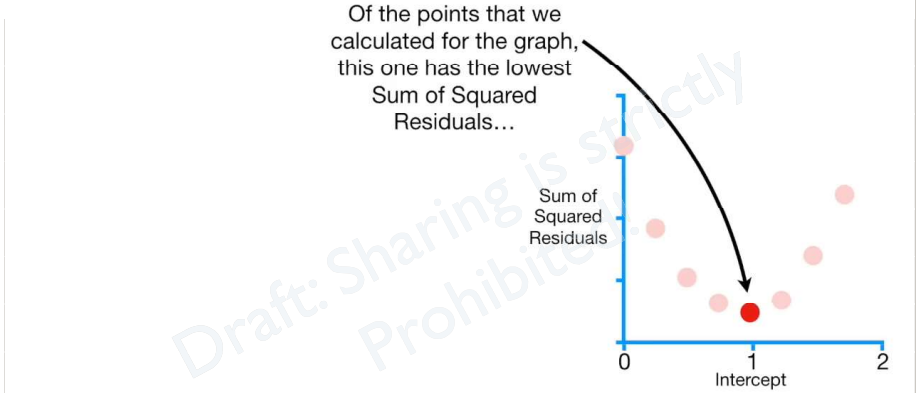



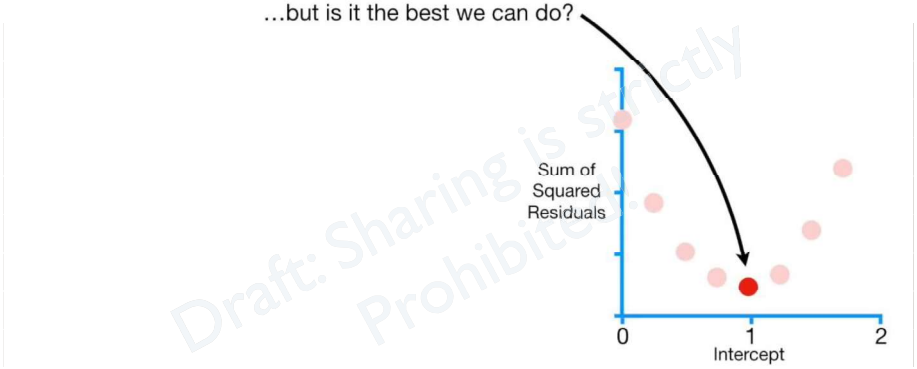



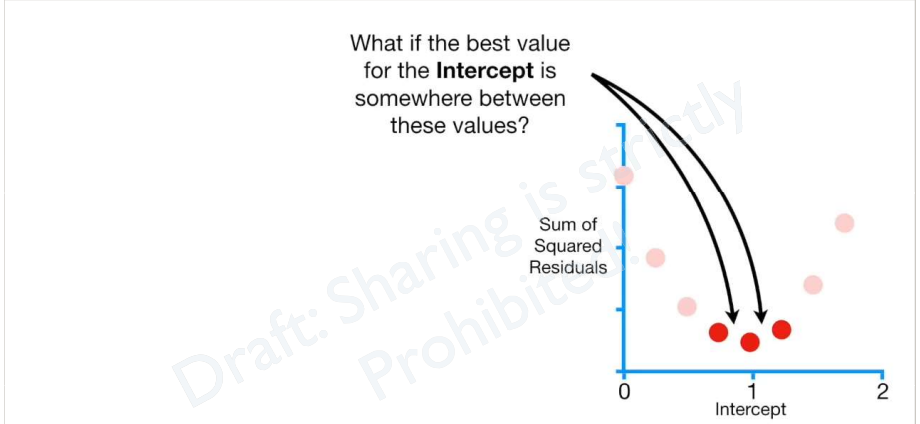


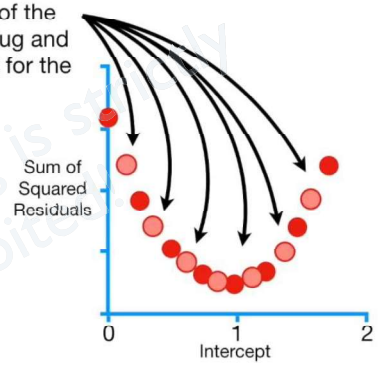



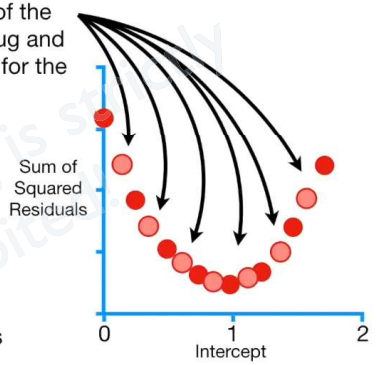



A slow and painful method for finding the minimal Sum of the Squared Residuals is to plug and chug a bunch more values for the **Intercept**.



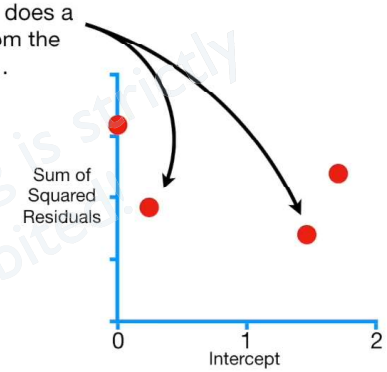

A slow and painful method for finding the minimal Sum of the Squared Residuals is to plug and chug a bunch more values for the **Intercept**.

Ugh.

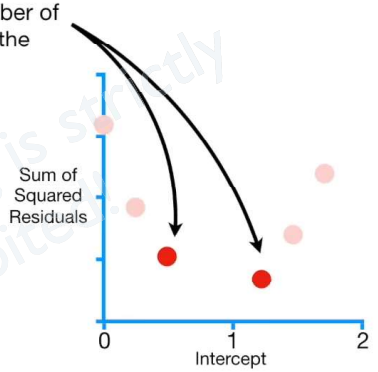



Don't despair! **Gradient Descent** is way more efficient!

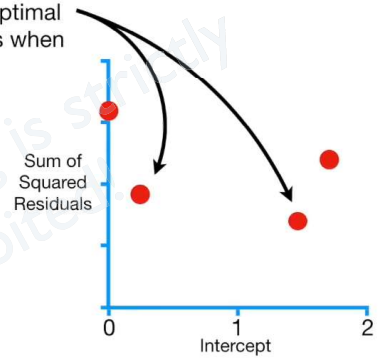
**Gradient Descent** only does a few calculations far from the optimal solution…



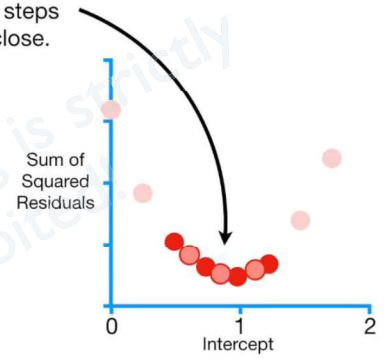

…and increases the number of calculations closer to the optimal value.




In other words, **Gradient Descent** identifies the optimal value by taking big steps when it is far away…

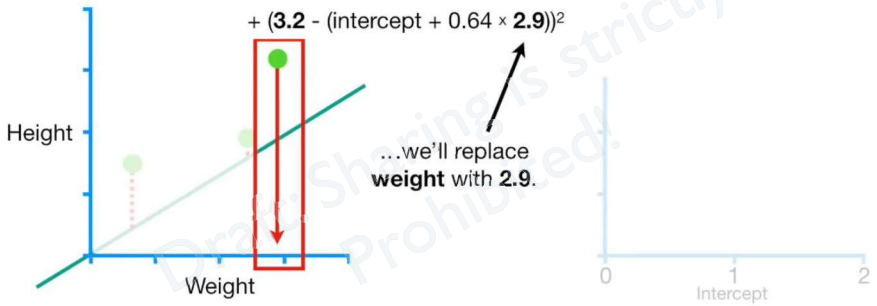



…and baby steps when it is close.

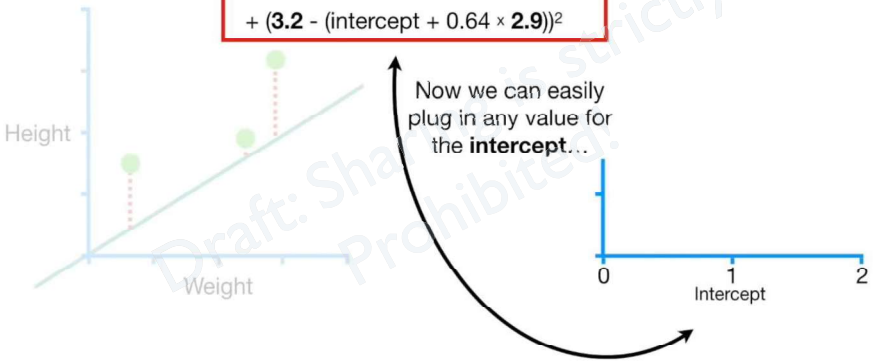


So let's get back to using **Gradient Descent** to find the optimal value for the **Intercept**, starting from a random value. In this case, the random value was **0**.

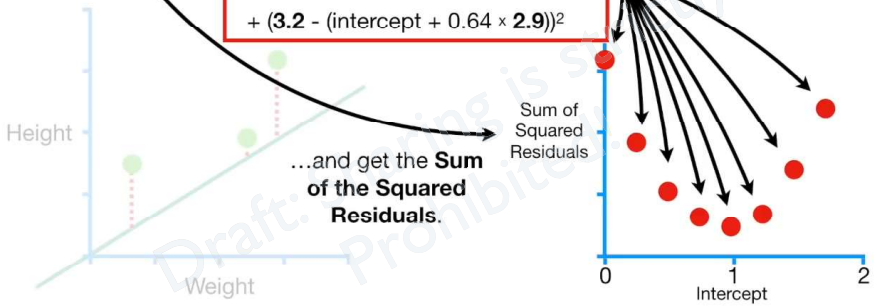

Sum of squared residuals = (**1.4** - (intercept + 0.64 × **0.5**))²
+ (**1.9** - (intercept + 0.64 × **2.3**))²
+ (**3.2** - (intercept + 0.64 × **2.9**))²



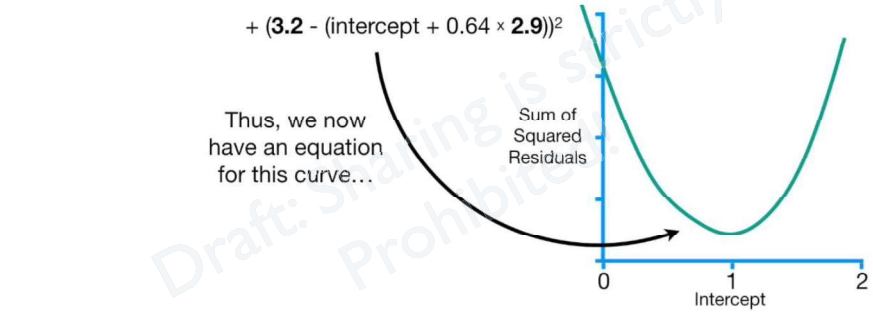

Sum of squared residuals = (**1.4** - (intercept + 0.64 × **0.5**))²
+ (**1.9** - (intercept + 0.64 × **2.3**))²
+ (**3.2** - (intercept + 0.64 × **2.9**))²



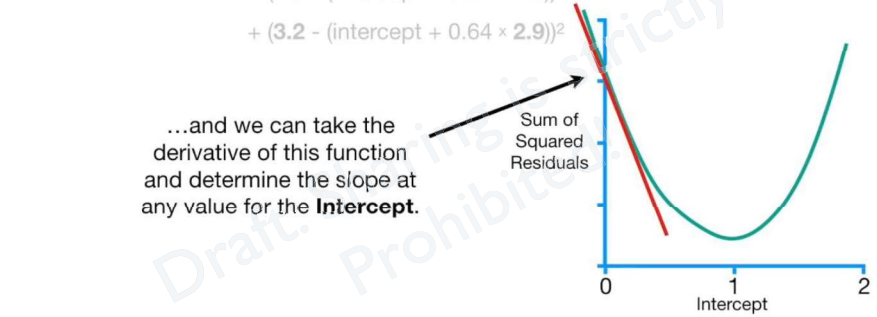

Sum of squared residuals = (**1.4** - (intercept + 0.64 × **0.5**))²
+ (**1.9** - (intercept + 0.64 × **2.3**))²
+ (**3.2** - (intercept + 0.64 × **2.9**))²




Sum of squared residuals = (**1.4** - (intercept + 0.64 × **0.5**))²
+ (**1.9** - (intercept + 0.64 × **2.3**))²
+ (**3.2** - (intercept + 0.64 × **2.9**))²

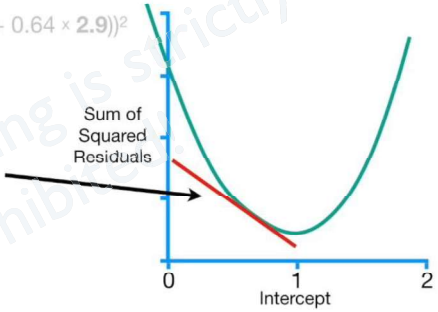



Sum of squared residuals = (**1.4** - (intercept + 0.64 × **0.5**))²
+ (**1.9** - (intercept + 0.64 × **2.3**))²
+ (**3.2** - (intercept + 0.64 × **2.9**))²



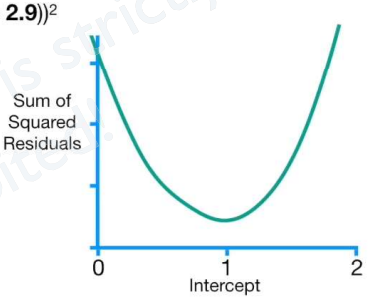
Sum of squared residuals = (**1.4** - (intercept + 0.64 × **0.5**))$^2$
+ (**1.9** - (intercept + 0.64 × **2.3**))$^2$
+ (**3.2** - (intercept + 0.64 × **2.9**))$^2$

…and we can take the derivative of this function and determine the slope at any value for the **Intercept**.




Sum of squared residuals = (**1.4** - (intercept + 0.64 × **0.5**))$^2$
+ (**1.9** - (intercept + 0.64 × **2.3**))$^2$
+ (**3.2** - (intercept + 0.64 × **2.9**))$^2$

So let's take the derivative of the Sum of the Squared Residuals with respect to the **Intercept**.

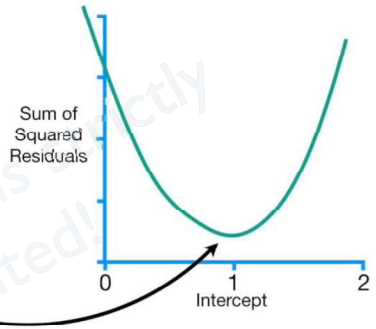



$\frac{d}{d\ intercept}$ Sum of squared residuals =
-2(**1.4** - (intercept + 0.64 × **0.5**)
+ -2(**1.9** - (intercept + 0.64 × **2.3**))
+ -2(**3.2** - (intercept + 0.64 × **2.9**))

Let's move the derivative up here so that it's not taking up half of the screen.

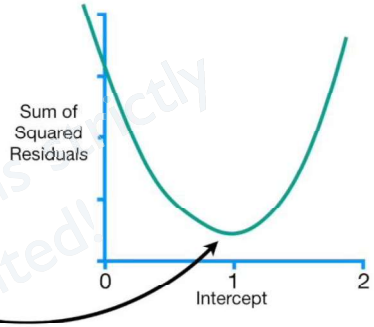
$\frac{d}{d\ intercept}$ Sum of squared residuals =
-2(**1.4** - (intercept + 0.64 × **0.5**)
+ -2(**1.9** - (intercept + 0.64 × **2.3**))
+ -2(**3.2** - (intercept + 0.64 × **2.9**))

Now that we have the derivative, **Gradient Descent** will use it to find where the Sum of Squared Residuals is lowest.



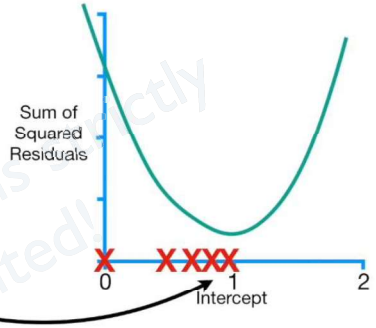

$\frac{d}{d\ intercept}$ Sum of squared residuals =
-2(**1.4** - (intercept + 0.64 × **0.5**)
+ -2(**1.9** - (intercept + 0.64 × **2.3**))
+ -2(**3.2** - (intercept + 0.64 × **2.9**))

**NOTE:** If we were using **Least Squares** to solve for the optimal value for the **Intercept**, we would simply find where the the slope of the curve = **0**.




$\frac{d}{d\ intercept}$ Sum of squared residuals =
-2(**1.4** - (intercept + 0.64 × **0.5**)
+ -2(**1.9** - (intercept + 0.64 × **2.3**))
+ -2(**3.2** - (intercept + 0.64 × **2.9**))

In contrast, **Gradient Descent** finds the minimum value by taking steps from an initial guess until it reaches the best value.



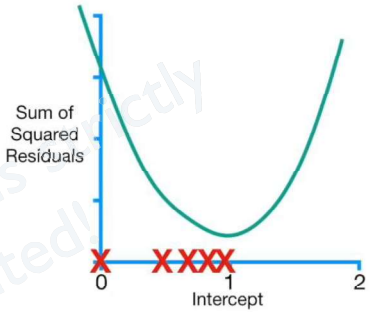
$\frac{d}{d\ intercept}$ Sum of squared residuals =

$-2(\mathbf{1.4} - (\text{intercept} + 0.64 \times \mathbf{0.5}))$

$+ -2(\mathbf{1.9} - (\text{intercept} + 0.64 \times \mathbf{2.3}))$

$+ -2(\mathbf{3.2} - (\text{intercept} + 0.64 \times \mathbf{2.9}))$

This makes **Gradient Descent** very useful when it is not possible to solve for where the derivative = **0**, and this is why **Gradient Descent** can be used in so many different situations.





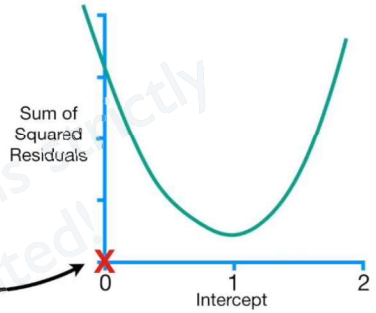

$\frac{d}{d\ intercept}$ Sum of squared residuals =

$-2(\mathbf{1.4} - (\text{intercept} + 0.64 \times \mathbf{0.5}))$

$+ -2(\mathbf{1.9} - (\text{intercept} + 0.64 \times \mathbf{2.3}))$

$+ -2(\mathbf{3.2} - (\text{intercept} + 0.64 \times \mathbf{2.9}))$

Remember, we started by setting the **Intercept** to a random number. In this case, that was **0**.





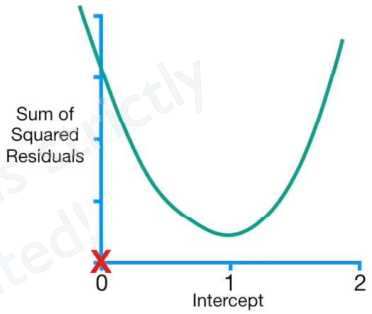

$\frac{d}{d\ intercept}$ Sum of squared residuals =

$-2(\mathbf{1.4} - (0 + 0.64 \times \mathbf{0.5}))$

$+ -2(\mathbf{1.9} - (0 + 0.64 \times \mathbf{2.3}))$

$+ -2(\mathbf{3.2} - (0 + 0.64 \times \mathbf{2.9}))$

So we plug **0** into the derivative...






$\frac{d}{d\ intercept}$ Sum of squared residuals =

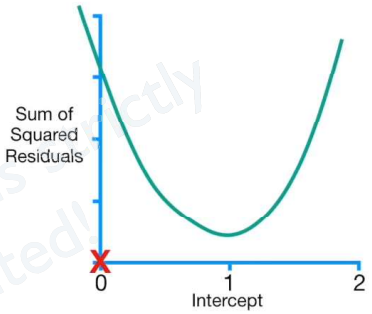
$-2(\mathbf{1.4} - (0 + 0.64 \times \mathbf{0.5}))$

$+ -2(\mathbf{1.9} - (0 + 0.64 \times \mathbf{2.3}))$

$+ -2(\mathbf{3.2} - (0 + 0.64 \times \mathbf{2.9}))$

$= -5.7$

...and we get **-5.7**.






$\frac{d}{d\ intercept}$ Sum of squared residuals =

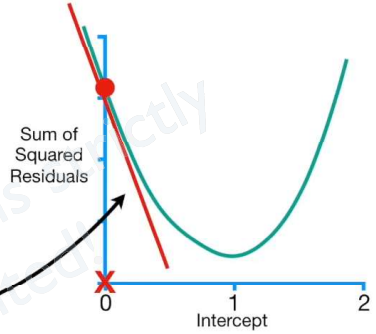
$-2(\mathbf{1.4} - (0 + 0.64 \times \mathbf{0.5}))$

$+ -2(\mathbf{1.9} - (0 + 0.64 \times \mathbf{2.3}))$

$+ -2(\mathbf{3.2} - (0 + 0.64 \times \mathbf{2.9}))$

$= -5.7$

So when the **Intercept** = **0**, the slope of the curve = **-5.7**.






$\frac{d}{d\ intercept}$ Sum of squared residuals =

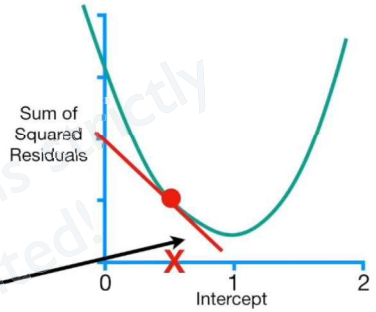
$-2(\mathbf{1.4} - (0 + 0.64 \times \mathbf{0.5}))$

$+ -2(\mathbf{1.9} - (0 + 0.64 \times \mathbf{2.3}))$

$+ -2(\mathbf{3.2} - (0 + 0.64 \times \mathbf{2.9}))$

$= -5.7$

**NOTE:** The closer we get to the optimal value for the **Intercept**, the closer the slope of the curve gets to **0**.



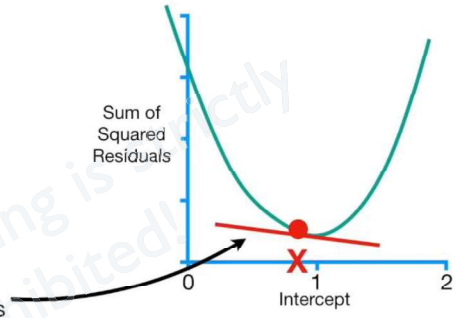
$\frac{d}{d\,intercept}$ Sum of squared residuals =

-2(**1.4** - (0 + 0.64 × **0.5**)

+ -2(**1.9** - (0 + 0.64 × **2.3**))

+ -2(**3.2** - (0 + 0.64 × **2.9**))

= -5.7




This means that when the slope of the curve is close to **0**...

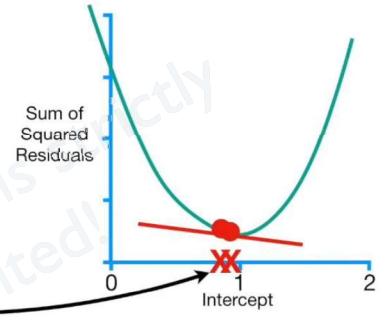
$\frac{d}{d\,intercept}$ Sum of squared residuals =

-2(**1.4** - (0 + 0.64 × **0.5**)

+ -2(**1.9** - (0 + 0.64 × **2.3**))

+ -2(**3.2** - (0 + 0.64 × **2.9**))

= -5.7




...then we should take baby steps, because we are close to the optimal value...

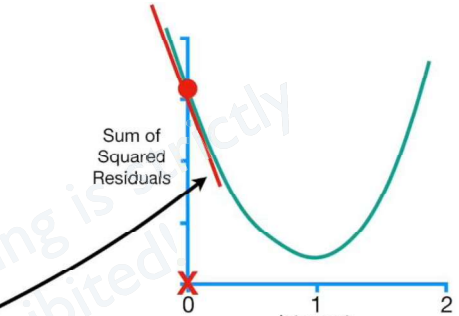
$\frac{d}{d\,intercept}$ Sum of squared residuals =

-2(**1.4** - (0 + 0.64 × **0.5**)

+ -2(**1.9** - (0 + 0.64 × **2.3**))

+ -2(**3.2** - (0 + 0.64 × **2.9**))

= -5.7




...and when the slope is far from **0**...

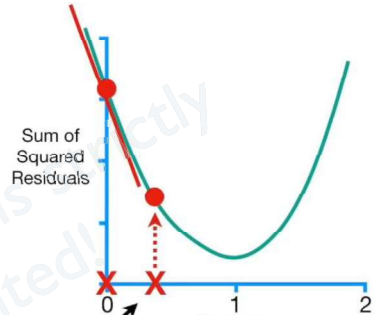
$\frac{d}{d\,intercept}$ Sum of squared residuals =

-2(**1.4** - (0 + 0.64 × **0.5**)

+ -2(**1.9** - (0 + 0.64 × **2.3**))

+ -2(**3.2** - (0 + 0.64 × **2.9**))

= -5.7




...then we should take big steps, because we are far from the optimal value.

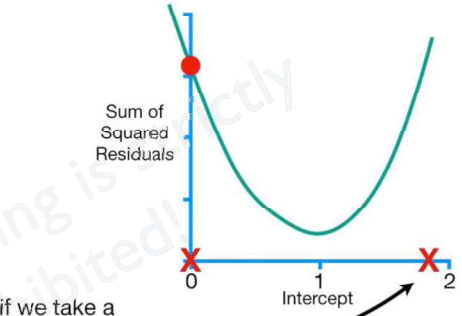
$\frac{d}{d\,intercept}$ Sum of squared residuals =

-2(**1.4** - (0 + 0.64 × **0.5**)

+ -2(**1.9** - (0 + 0.64 × **2.3**))

+ -2(**3.2** - (0 + 0.64 × **2.9**))

= -5.7




However, if we take a super huge step...

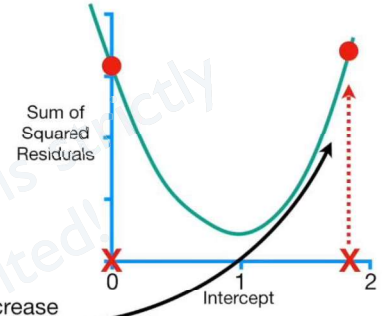
$\frac{d}{d\,intercept}$ Sum of squared residuals =

-2(**1.4** - (0 + 0.64 × **0.5**)

+ -2(**1.9** - (0 + 0.64 × **2.3**))

+ -2(**3.2** - (0 + 0.64 × **2.9**))

= -5.7




...then we would increase the Sum of the Squared Residuals!

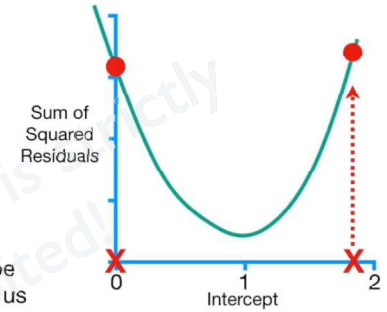
$\frac{d}{d\,intercept}$ Sum of squared residuals =

$-2(\mathbf{1.4} - (0 + 0.64 \times \mathbf{0.5})$

$+ -2(\mathbf{1.9} - (0 + 0.64 \times \mathbf{2.3}))$

$+ -2(\mathbf{3.2} - (0 + 0.64 \times \mathbf{2.9}))$

$= -5.7$




So the size of the step should be related to the slope, since it tells us if we should take a baby step or a big step, but we need to make sure the big step is not too big.



$\frac{d}{d\,intercept}$ Sum of squared residuals =

$-2(\mathbf{1.4} - (0 + 0.64 \times \mathbf{0.5})$

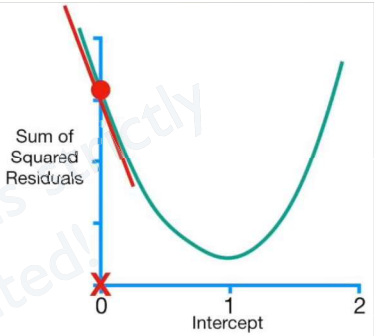
$+ -2(\mathbf{1.9} - (0 + 0.64 \times \mathbf{2.3}))$

$+ -2(\mathbf{3.2} - (0 + 0.64 \times \mathbf{2.9}))$

$= -5.7$

**Step Size** = -5.7

**Gradient Descent** determines the **Step Size** by multiplying the **slope**…






$\frac{d}{d\,intercept}$ Sum of squared residuals =

$-2(\mathbf{1.4} - (0 + 0.64 \times \mathbf{0.5})$

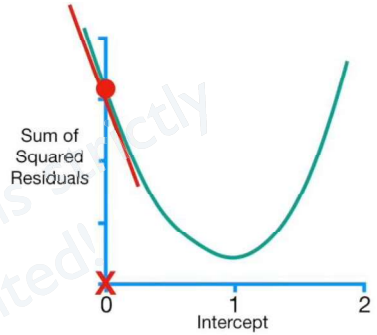
$+ -2(\mathbf{1.9} - (0 + 0.64 \times \mathbf{2.3}))$

$+ -2(\mathbf{3.2} - (0 + 0.64 \times \mathbf{2.9}))$

$= -5.7$

**Step Size** = -5.7 × 0.1

…by a small number called **The Learning Rate**.






$\frac{d}{d\,intercept}$ Sum of squared residuals =

$-2(\mathbf{1.4} - (0 + 0.64 \times \mathbf{0.5})$

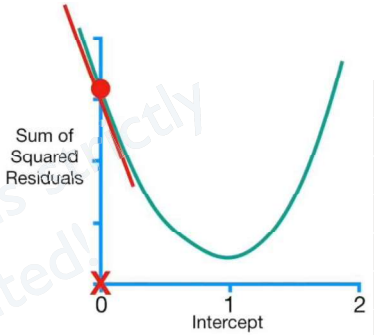
$+ -2(\mathbf{1.9} - (0 + 0.64 \times \mathbf{2.3}))$

$+ -2(\mathbf{3.2} - (0 + 0.64 \times \mathbf{2.9}))$

$= -5.7$

**Step Size** = -5.7 × 0.1 = -0.57

When the **Intercept** = **0**, the **Step Size** = **-0.57**.






$\frac{d}{d\,intercept}$ Sum of squared residuals =

$-2(\mathbf{1.4} - (0 + 0.64 \times \mathbf{0.5})$

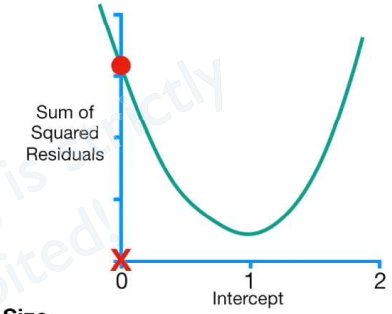
$+ -2(\mathbf{1.9} - (0 + 0.64 \times \mathbf{2.3}))$

$+ -2(\mathbf{3.2} - (0 + 0.64 \times \mathbf{2.9}))$

$= -5.7$

**Step Size** = -5.7 × 0.1 = -0.57

**New Intercept** = With the **Step Size**, we can calculate a **New Intercept**.






$\frac{d}{d\,intercept}$ Sum of squared residuals =

$-2(\mathbf{1.4} - (0 + 0.64 \times \mathbf{0.5})$

$+ -2(\mathbf{1.9} - (0 + 0.64 \times \mathbf{2.3}))$

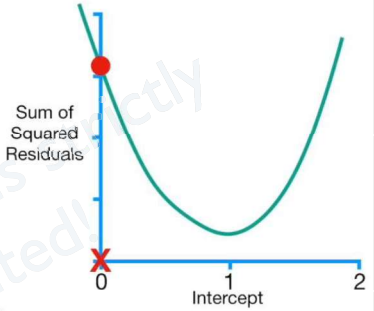
$+ -2(\mathbf{3.2} - (0 + 0.64 \times \mathbf{2.9}))$

$= -5.7$

**Step Size** = -5.7 × 0.1 = -0.57

**New Intercept** = **Old Intercept** - **Step Size**

…minus the **Step Size**.



$\frac{d}{d\ intercept}$ Sum of squared residuals =

$-2(\mathbf{1.4} - (0 + 0.64 \times \mathbf{0.5})$

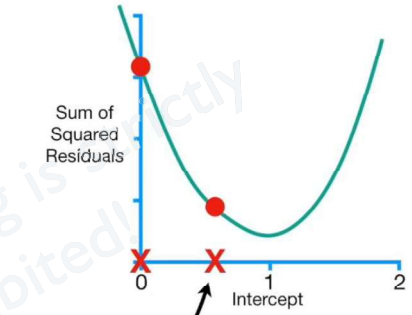
$+ -2(\mathbf{1.9} - (0 + 0.64 \times \mathbf{2.3}))$

$+ -2(\mathbf{3.2} - (0 + 0.64 \times \mathbf{2.9}))$

$= -5.7$

**Step Size** $= -5.7 \times 0.1 = -0.57$

**New Intercept** $= 0 - (-0.57) = \mathbf{0.57}$




...and the the **New Intercept** = **0.57**.

$\frac{d}{d\ intercept}$ Sum of squared residuals =

$-2(\mathbf{1.4} - (0 + 0.64 \times \mathbf{0.5})$

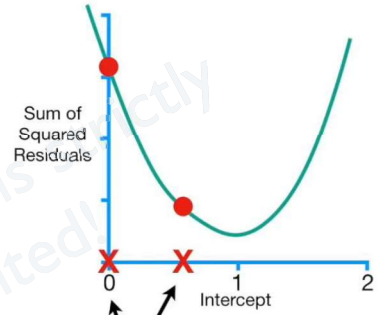
$+ -2(\mathbf{1.9} - (0 + 0.64 \times \mathbf{2.3}))$

$+ -2(\mathbf{3.2} - (0 + 0.64 \times \mathbf{2.9}))$

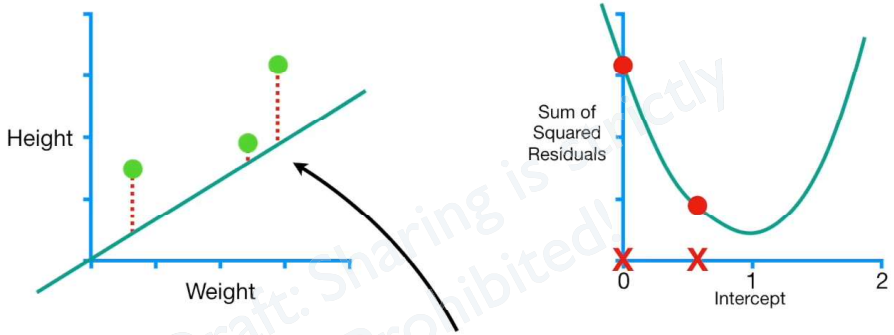
$= -5.7$

**Step Size** $= -5.7 \times 0.1 = -0.57$

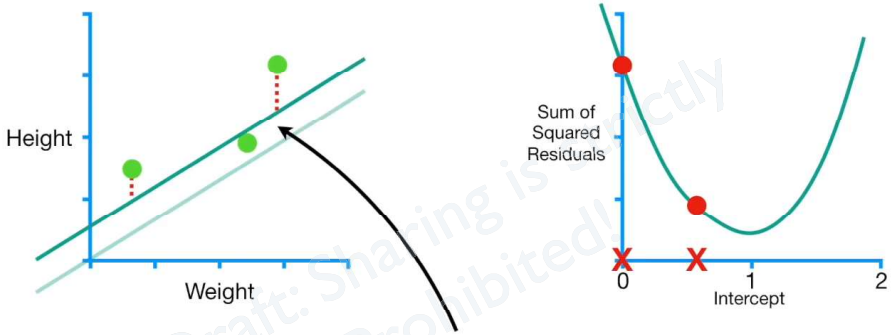
**New Intercept** $= 0 - (-0.57) = \mathbf{0.57}$



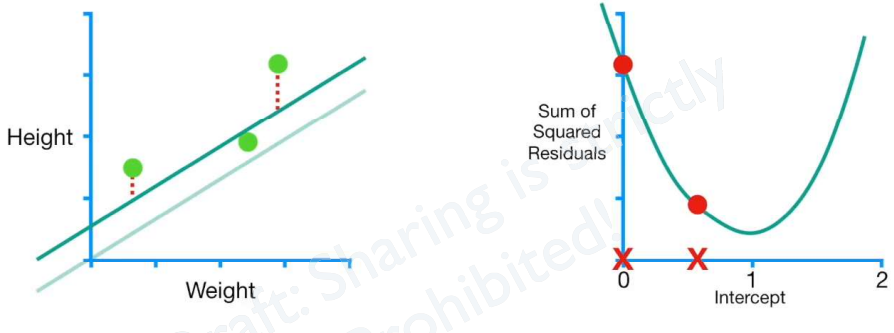

In one big step, we moved much closer to the optimal value for the **Intercept**.




Going back to the original data and the original line, with the **Intercept** = **0**...




...we can see how much the residuals shrink when the **Intercept** = **0.57**.



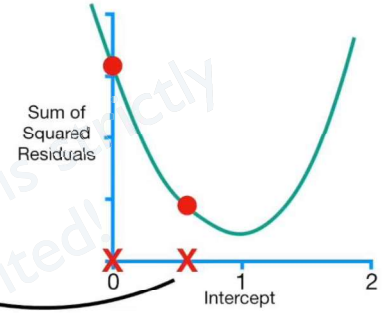

Now let's take another step closer to the optimal value for the **Intercept**.

$\frac{d}{d\ intercept}$ Sum of squared residuals =

$-2(\mathbf{1.4} - (0.57 + 0.64 \times \mathbf{0.5})$

$+ -2(\mathbf{1.9} - (0.57 + 0.64 \times \mathbf{2.3}))$

$+ -2(\mathbf{3.2} - (0.57 + 0.64 \times \mathbf{2.9}))$




To take another step, we go back to the derivative and plug in the **New Intercept** (**0.57**)...

$\frac{d}{d\ intercept}$ Sum of squared residuals =

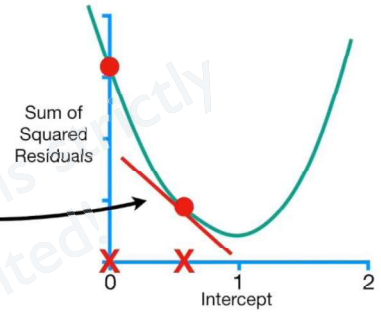
-2(**1.4** - (0.57 + 0.64 × **0.5**)

+ -2(**1.9** - (0.57 + 0.64 × **2.3**))

+ -2(**3.2** - (0.57 + 0.64 × **2.9**))

= -2.3

…and that tells us the slope of the curve = **-2.3**.




$\frac{d}{d\ intercept}$ Sum of squared residuals =

-2(**1.4** - (0.57 + 0.64 × **0.5**)

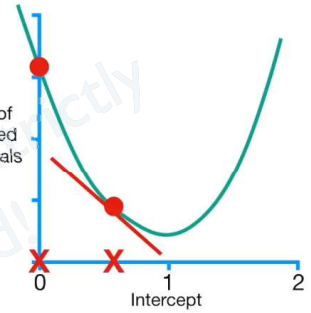
+ -2(**1.9** - (0.57 + 0.64 × **2.3**))

+ -2(**3.2** - (0.57 + 0.64 × **2.9**))

= -2.3

**Step Size** = **Slope** × **Learning Rate**

Now let's calculate the **Step Size**…




$\frac{d}{d\ intercept}$ Sum of squared residuals =

-2(**1.4** - (0.57 + 0.64 × **0.5**)

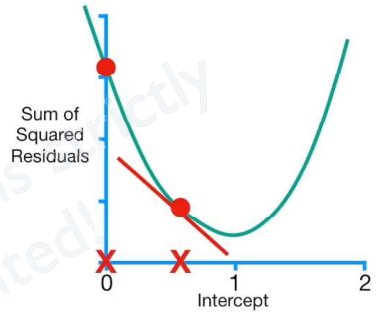
+ -2(**1.9** - (0.57 + 0.64 × **2.3**))

+ -2(**3.2** - (0.57 + 0.64 × **2.9**))

= -2.3

**Step Size** = -2.3 × 0.1 = **-0.23**

Ultimately, the **Step Size** is **-0.23**…




$\frac{d}{d\ intercept}$ Sum of squared residuals =

-2(**1.4** - (0.57 + 0.64 × **0.5**)

+ -2(**1.9** - (0.57 + 0.64 × **2.3**))

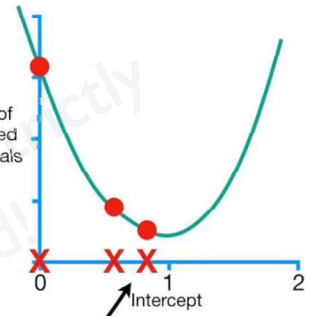
+ -2(**3.2** - (0.57 + 0.64 × **2.9**))

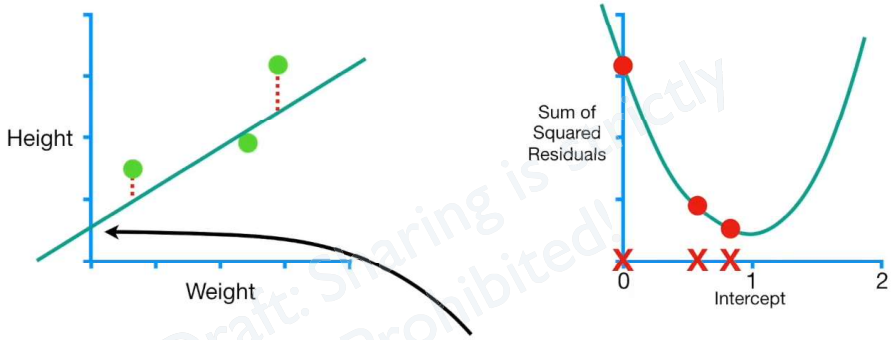
= -2.3

**Step Size** = -2.3 × 0.1 = **-0.23**

**New Intercept** = 0.57 - (-0.23) = **0.8**

…and the **New Intercept** = **0.8**

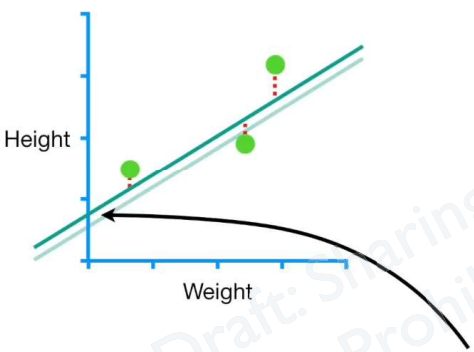


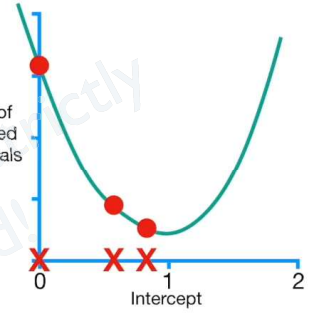





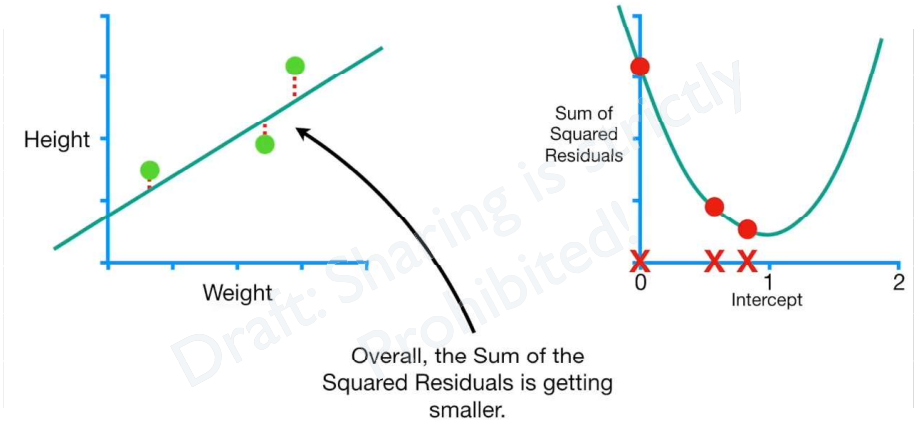


Now we can compare the residuals when the **Intercept** = **0.57**…




…to when the **Intercept** = **0.8**

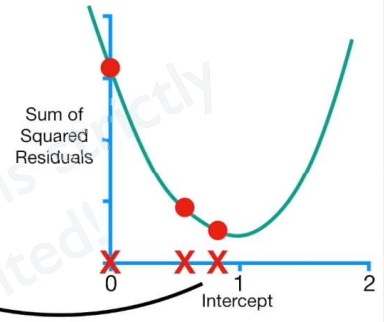



$\frac{d}{d\ intercept}$ Sum of squared residuals =

$-2(\mathbf{1.4} - (0.8 + 0.64 \times \mathbf{0.5})$

$+ -2(\mathbf{1.9} - (0.8 + 0.64 \times \mathbf{2.3}))$

$+ -2(\mathbf{3.2} - (0.8 + 0.64 \times \mathbf{2.9}))$



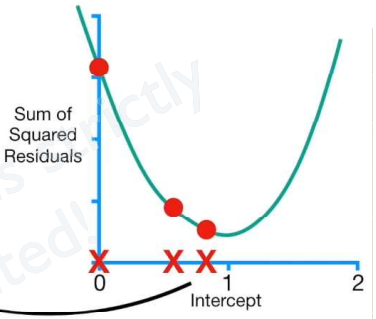

Now let's calculate the derivative at the **New Intercept** (**0.8**)...

$\frac{d}{d\ intercept}$ Sum of squared residuals =

$-2(\mathbf{1.4} - (0.8 + 0.64 \times \mathbf{0.5})$

$+ -2(\mathbf{1.9} - (0.8 + 0.64 \times \mathbf{2.3}))$

$+ -2(\mathbf{3.2} - (0.8 + 0.64 \times \mathbf{2.9}))$




Now let's calculate the derivative at the **New Intercept** (**0.8**)...

$\frac{d}{d\ intercept}$ Sum of squared residuals =

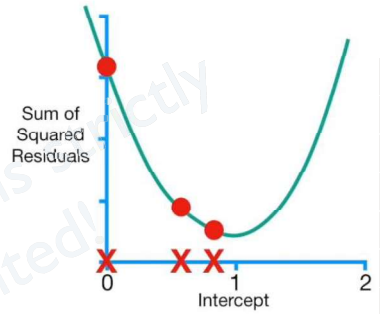
$-2(\mathbf{1.4} - (0.8 + 0.64 \times \mathbf{0.5})$

$+ -2(\mathbf{1.9} - (0.8 + 0.64 \times \mathbf{2.3}))$

$+ -2(\mathbf{3.2} - (0.8 + 0.64 \times \mathbf{2.9}))$

$= \mathbf{-0.9}$

...and we get **-0.9**.




$\frac{d}{d\ intercept}$ Sum of squared residuals =

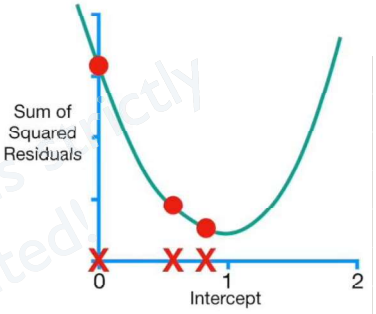
$-2(\mathbf{1.4} - (0.8 + 0.64 \times \mathbf{0.5})$

$+ -2(\mathbf{1.9} - (0.8 + 0.64 \times \mathbf{2.3}))$

$+ -2(\mathbf{3.2} - (0.8 + 0.64 \times \mathbf{2.9}))$

$= \mathbf{-0.9}$

**Step Size** = $-0.9 \times 0.1 = \mathbf{-0.09}$




The **Step Size** = **-0.09**...

$\frac{d}{d\ intercept}$ Sum of squared residuals =

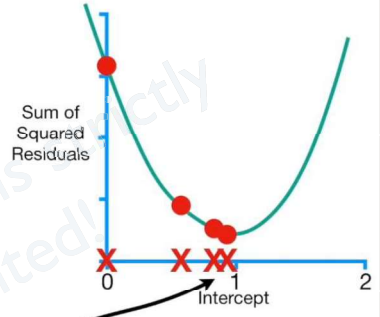
$-2(\mathbf{1.4} - (0.8 + 0.64 \times \mathbf{0.5})$

$+ -2(\mathbf{1.9} - (0.8 + 0.64 \times \mathbf{2.3}))$

$+ -2(\mathbf{3.2} - (0.8 + 0.64 \times \mathbf{2.9}))$

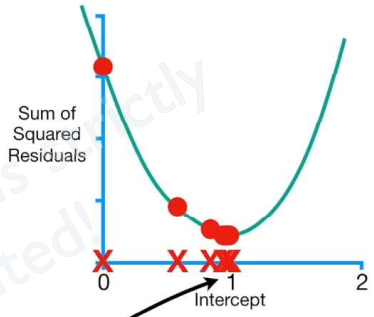
$= \mathbf{-0.9}$

**Step Size** = $-0.9 \times 0.1 = \mathbf{-0.09}$




**New Intercept** = 0.8 - (-0.09) = **0.89**

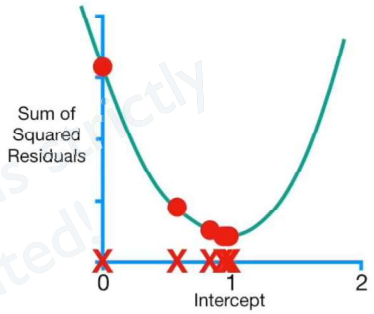
...and the **New Intercept** = **0.89**




Notice how each step gets smaller and smaller the closer we get to the bottom of the curve.

**Gradient Descent** stops when the **Step Size** is **Very Close To 0**.

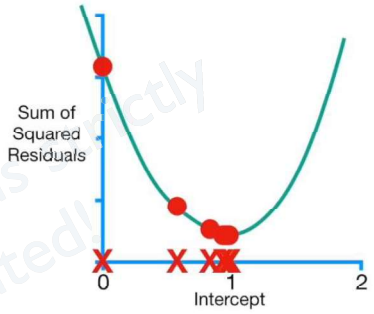
**Step Size** = **Slope** × **Learning Rate**



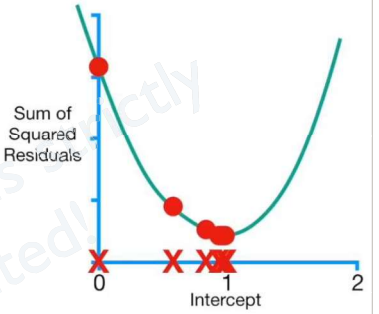

After 6 steps, the **Gradient Descent** estimate for the **Intercept** is **0.95**.

**NOTE:** The **Least Squares** estimate for the intercept is also **0.95**.

So we know that **Gradient Descent** has done its job, but without comparing its solution to a gold standard, how does **Gradient Descent** know to stop taking steps?

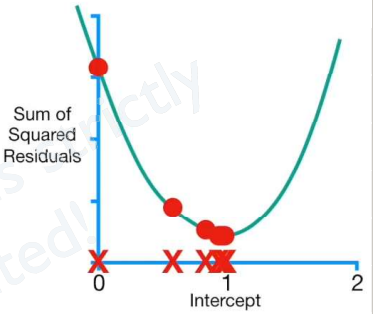



After 6 steps, the **Gradient Descent** estimate for the **Intercept** is **0.95**.



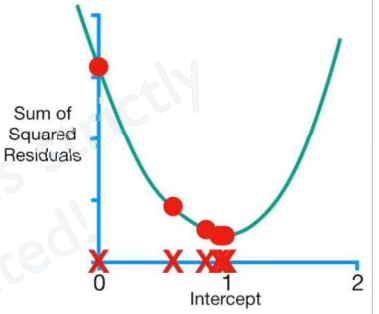

After 6 steps, the **Gradient Descent** estimate for the **Intercept** is **0.95**.

**NOTE:** The **Least Squares** estimate for the intercept is also **0.95**.



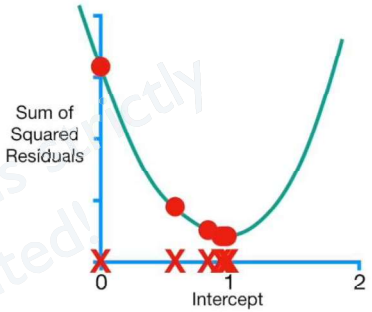

**Gradient Descent** stops when the **Step Size** is **Very Close To 0**.

**Step Size** = **Slope** × **Learning Rate**



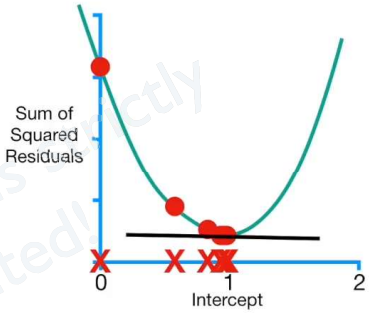
**Gradient Descent** stops when the **Step Size** is **Very Close To 0**.

**Step Size** = **Slope** × **Learning Rate**



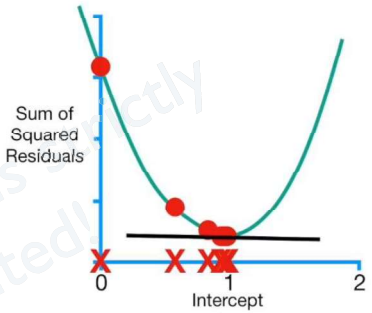

Then we would plug in **0.009** for the **Slope** and **0.1** for the **Learning Rate**..

**Step Size** = **0.009** × **0.1**

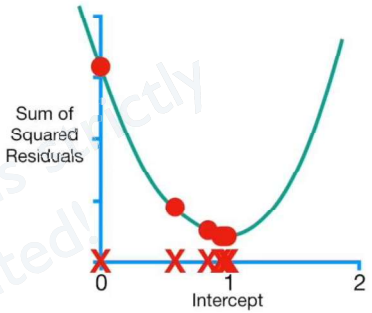



...and get **0.0009**, which is smaller than **0.001**, so **Gradient Descent** would stop.

**Step Size** = **0.009** × **0.1** = 0.0009

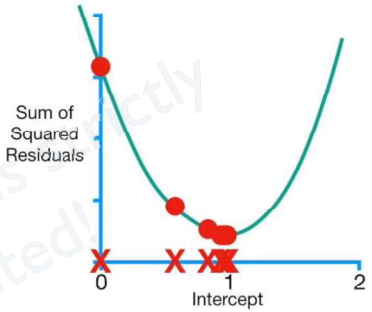



That said, **Gradient Descent** also includes a limit on the number of steps it will take before giving up.

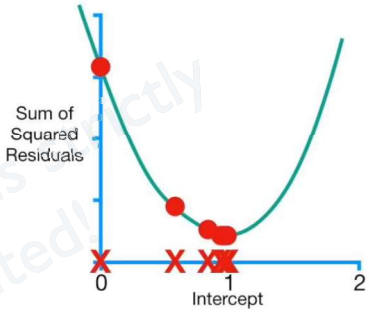



That said, **Gradient Descent** also includes a limit on the number of steps it will take before giving up.

In practice, the **Maximum Number of Steps** = **1,000** or greater.

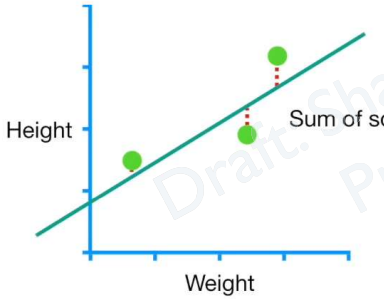



So, even if the **Step Size** is large, if there have been more than the **Maximum Number of Steps**, **Gradient Descent** will stop.



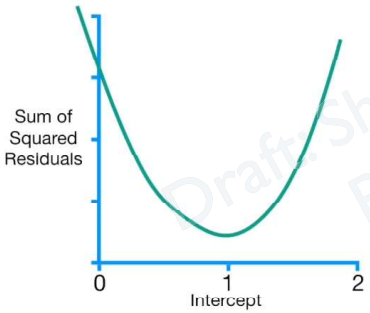
OK, let's review what we've learned so far…

The first thing we did is decide to use the Sum of the Squared Residuals as the **Loss Function** to evaluate how well a line fits the data…



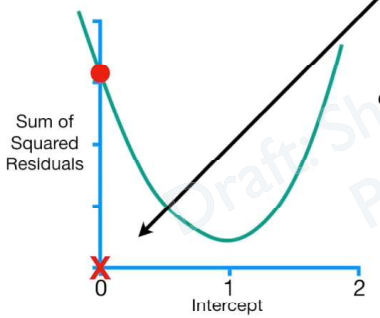

Sum of squared residuals = (**1.4** - (intercept + 0.64 × **0.5**))$^2$
+ (**1.9** - (intercept + 0.64 × **2.3**))$^2$
+ (**3.2** - (intercept + 0.64 × **2.9**))$^2$

…then we took the derivative of the Sum of the Squared Residuals. In other words, we took the derivative of the **Loss Function**…



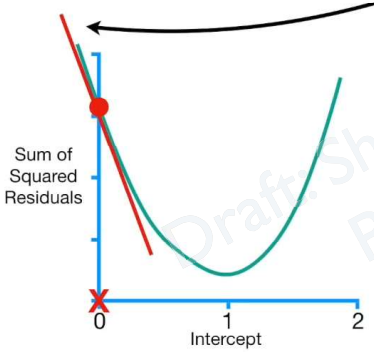

$\frac{d}{d\,intercept}$ Sum of squared residuals =
-2(**1.4** - (intercept + 0.64 × **0.5**)
+ -2(**1.9** - (intercept + 0.64 × **2.3**))
+ -2(**3.2** - (intercept + 0.64 × **2.9**))

…then we picked a random value for the **Intercept**, in this case we set the **Intercept** = **0**…




$\frac{d}{d\,intercept}$ Sum of squared residuals =
-2(**1.4** - (intercept + 0.64 × **0.5**)
+ -2(**1.9** - (intercept + 0.64 × **2.3**))
+ -2(**3.2** - (intercept + 0.64 × **2.9**))

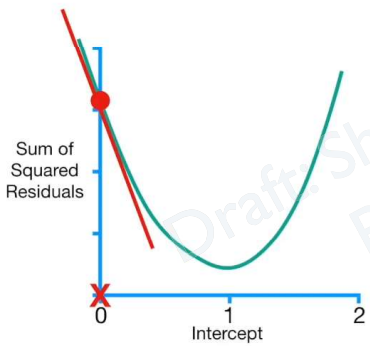
…then we calculated the derivative when the **Intercept** = **0**…



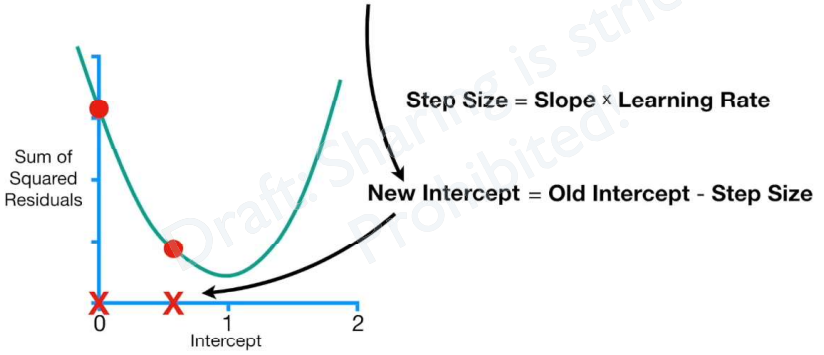

$\frac{d}{d\,intercept}$ Sum of squared residuals =
-2(**1.4** - (intercept + 0.64 × **0.5**)
+ -2(**1.9** - (intercept + 0.64 × **2.3**))
+ -2(**3.2** - (intercept + 0.64 × **2.9**))

…plugged that slope into the **Step Size** calculation…

**Step Size** = **Slope** × **Learning Rate**



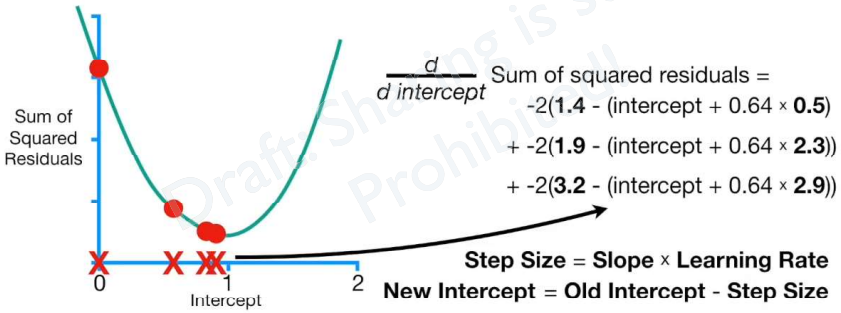
…then calculated the **New Intercept**, the difference between the **Old Intercept** and the **Step Size**.




**Step Size** = **Slope** × **Learning Rate**

**New Intercept** = **Old Intercept** - **Step Size**

Lastly, we plugged the **New Intercept** into the derivative and repeated everything until **Step Size** was close to **0**.




$\frac{d}{d\ intercept}$ Sum of squared residuals =

$-2(\mathbf{1.4} - (\text{intercept} + 0.64 \times \mathbf{0.5})$

$+ -2(\mathbf{1.9} - (\text{intercept} + 0.64 \times \mathbf{2.3}))$

$+ -2(\mathbf{3.2} - (\text{intercept} + 0.64 \times \mathbf{2.9}))$

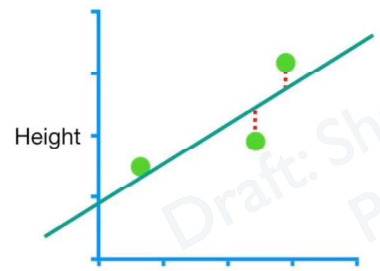
**Step Size** = **Slope** × **Learning Rate**

**New Intercept** = **Old Intercept** - **Step Size**

Sum of squared residuals = $(\mathbf{1.4} - (\text{intercept} + \text{slope} \times \mathbf{0.5}))^2$

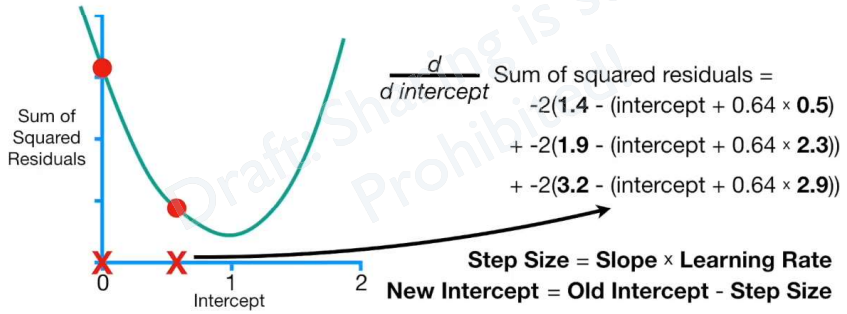
$+ (\mathbf{1.9} - (\text{intercept} + \text{slope} \times \mathbf{2.3}))^2$

$+ (\mathbf{3.2} - (\text{intercept} + \text{slope} \times \mathbf{2.9}))^2$




Just like before, we will use the Sum of the Squared Residuals as the **Loss Function**

Lastly, we plugged the **New Intercept** into the derivative and repeated everything until **Step Size** was close to **0**.




$\frac{d}{d\ intercept}$ Sum of squared residuals =

$-2(\mathbf{1.4} - (\text{intercept} + 0.64 \times \mathbf{0.5})$

$+ -2(\mathbf{1.9} - (\text{intercept} + 0.64 \times \mathbf{2.3}))$

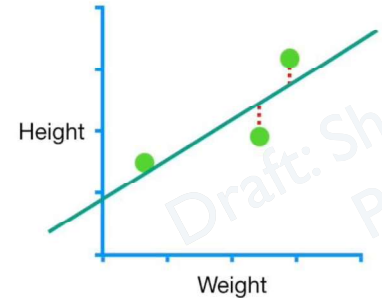
$+ -2(\mathbf{3.2} - (\text{intercept} + 0.64 \times \mathbf{2.9}))$

**Step Size** = **Slope** × **Learning Rate**

**New Intercept** = **Old Intercept** - **Step Size**

**Predicted Height** = intercept + slope × **Weight**

…let's talk abut how to estimate the **Intercept** *and* the **Slope**.

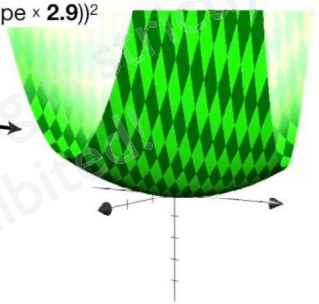



Sum of squared residuals = $(\mathbf{1.4} - (\text{intercept} + \text{slope} \times \mathbf{0.5}))^2$

$+ (\mathbf{1.9} - (\text{intercept} + \text{slope} \times \mathbf{2.3}))^2$

$+ (\mathbf{3.2} - (\text{intercept} + \text{slope} \times \mathbf{2.9}))^2$

This is a 3-D graph of the **Loss Function** for different values for the **Intercept** and the **Slope**



Sum of squared residuals = (**1.4** - (intercept + slope × **0.5**))$^2$

+ (**1.9** - (intercept + slope × **2.3**))$^2$

+ (**3.2** - (intercept + slope × **2.9**))$^2$

This is a 3-D graph of the **Loss Function** for different values for the **Intercept** and the **Slope**

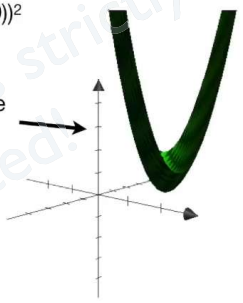


Sum of squared residuals = (**1.4** - (intercept + slope × **0.5**))$^2$

+ (**1.9** - (intercept + slope × **2.3**))$^2$

+ (**3.2** - (intercept + slope × **2.9**))$^2$

This axis is the Sum of the Squared Residuals…

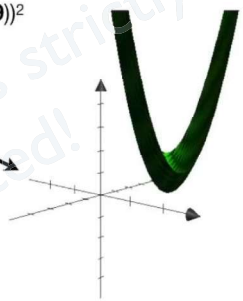


Sum of squared residuals = (**1.4** - (intercept + slope × **0.5**))$^2$

+ (**1.9** - (intercept + slope × **2.3**))$^2$

+ (**3.2** - (intercept + slope × **2.9**))$^2$

…this axis represents different values for the **Slope**…

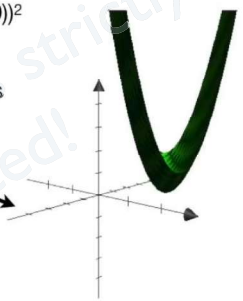


Sum of squared residuals = (**1.4** - (intercept + slope × **0.5**))$^2$

+ (**1.9** - (intercept + slope × **2.3**))$^2$

+ (**3.2** - (intercept + slope × **2.9**))$^2$

…and this axis represents different values for the **Intercept**.



Sum of squared residuals = (**1.4** - (intercept + slope × **0.5**))$^2$

+ (**1.9** - (intercept + slope × **2.3**))$^2$

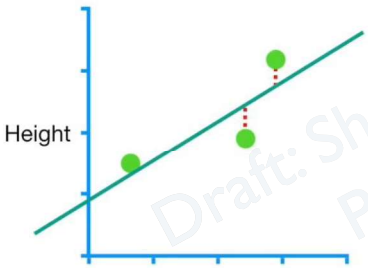
+ (**3.2** - (intercept + slope × **2.9**))$^2$

We want to find the values for the **Intercept** and **Slope** that give us the minimum Sum of the Squared Residuals.

Sum of squared residuals = (**1.4** - (intercept + slope × **0.5**))$^2$

+ (**1.9** - (intercept + slope × **2.3**))$^2$

+ (**3.2** - (intercept + slope × **2.9**))$^2$




So, just like before, we need to take the derivative of this function…

$$\frac{d}{d\,intercept}\text{ Sum of squared residuals} = -2(\mathbf{1.4} - (\text{intercept} + \text{slope} \times \mathbf{0.5})) + -2(\mathbf{1.9} - (\text{intercept} + \text{slope} \times \mathbf{2.3})) + -2(\mathbf{3.2} - (\text{intercept} + \text{slope} \times \mathbf{2.9}))$$

Here's the derivative of the Sum of the Squared Residuals with respect to the **Intercept**…

$$\frac{d}{d\,slope}\text{ Sum of squared residuals} = -2 \times \mathbf{0.5}(\mathbf{1.4} - (\text{intercept} + \text{slope} \times \mathbf{0.5})) + -2 \times \mathbf{2.9}(\mathbf{3.2} - (\text{intercept} + \text{slope} \times \mathbf{2.9})) + -2 \times \mathbf{2.3}(\mathbf{1.9} - (\text{intercept} + \text{slope} \times \mathbf{2.3}))$$

…and here's the derivative with respect to the **Slope**.

**NOTE:** When you have two or more derivatives of the same function, they are called a **Gradient**.

We will use this **Gradient** to **descend** to lowest point in the **Loss Function**, which, in this case, is the Sum of the Squared Residuals…

…thus, this is why this algorithm is called **Gradient Descent**!

Just like before, we will start by picking a random number for the **Intercept**. In this case we'll set the **Intercept** = **0**…

…and we'll pick a random number for the **Slope**. In this case we'll set the **Slope** = **1**.

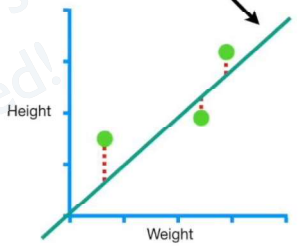
$\frac{d}{d\,intercept}$ Sum of squared residuals =

$-2(\mathbf{1.4} - (\text{intercept} + \text{slope} \times \mathbf{0.5}))$

$+ -2(\mathbf{1.9} - (\text{intercept} + \text{slope} \times \mathbf{2.3}))$

$+ -2(\mathbf{3.2} - (\text{intercept} + \text{slope} \times \mathbf{2.9}))$

Thus, this line, with **Intercept** = **0** and **Slope** = **1**, is where we will start.




$\frac{d}{d\,slope}$ Sum of squared residuals =

$-2 \times \mathbf{0.5}(\mathbf{1.4} - (\text{intercept} + \text{slope} \times \mathbf{0.5}))$

$+ -2 \times \mathbf{2.9}(\mathbf{3.2} - (\text{intercept} + \text{slope} \times \mathbf{2.9}))$

$+ -2 \times \mathbf{2.3}(\mathbf{1.9} - (\text{intercept} + \text{slope} \times \mathbf{2.3}))$



$\frac{d}{d\,intercept}$ Sum of squared residuals =

$-2(\mathbf{1.4} - (\text{intercept} + \text{slope} \times \mathbf{0.5}))$

$+ -2(\mathbf{1.9} - (\text{intercept} + \text{slope} \times \mathbf{2.3}))$

$+ -2(\mathbf{3.2} - (\text{intercept} + \text{slope} \times \mathbf{2.9}))$

Now let's plug in **0** for the **Intercept** and **1** for the **Slope**...

$\frac{d}{d\,slope}$ Sum of squared residuals =

$-2 \times \mathbf{0.5}(\mathbf{1.4} - (\text{intercept} + \text{slope} \times \mathbf{0.5}))$

$+ -2 \times \mathbf{2.9}(\mathbf{3.2} - (\text{intercept} + \text{slope} \times \mathbf{2.9}))$

$+ -2 \times \mathbf{2.3}(\mathbf{1.9} - (\text{intercept} + \text{slope} \times \mathbf{2.3}))$



$\frac{d}{d\,intercept}$ Sum of squared residuals =

$-2(\mathbf{1.4} - (0 + 1 \times \mathbf{0.5}))$

$+ -2(\mathbf{1.9} - (0 + 1 \times \mathbf{2.3}))$

$+ -2(\mathbf{3.2} - (0 + 1 \times \mathbf{2.9})) = \mathbf{-1.6}$

...and that gives us two **Slopes**...

$\frac{d}{d\,slope}$ Sum of squared residuals =

$-2 \times \mathbf{0.5}(\mathbf{1.4} - (0 + 1 \times \mathbf{0.5}))$

$+ -2 \times \mathbf{2.9}(\mathbf{3.2} - (0 + 1 \times \mathbf{2.9}))$

$+ -2 \times \mathbf{2.3}(\mathbf{1.9} - (0 + 1 \times \mathbf{2.3})) = \mathbf{-0.8}$



$\frac{d}{d\,intercept}$ Sum of squared residuals =

$-2(\mathbf{1.4} - (0 + 1 \times \mathbf{0.5}))$

$+ -2(\mathbf{1.9} - (0 + 1 \times \mathbf{2.3}))$

$+ -2(\mathbf{3.2} - (0 + 1 \times \mathbf{2.9})) = \mathbf{-1.6}$

$\mathbf{Step\ Size}_{Intercept} = \mathbf{Slope} \times \mathbf{Learning\ Rate}$

...now we plug the **Slopes** into the **Step Size** formulas...

$\frac{d}{d\,slope}$ Sum of squared residuals =

$-2 \times \mathbf{0.5}(\mathbf{1.4} - (0 + 1 \times \mathbf{0.5}))$

$+ -2 \times \mathbf{2.9}(\mathbf{3.2} - (0 + 1 \times \mathbf{2.9}))$

$+ -2 \times \mathbf{2.3}(\mathbf{1.9} - (0 + 1 \times \mathbf{2.3})) = \mathbf{-0.8}$

$\mathbf{Step\ Size}_{Slope} = \mathbf{Slope} \times \mathbf{Learning\ Rate}$



$\frac{d}{d\,intercept}$ Sum of squared residuals =

$-2(\mathbf{1.4} - (0 + 1 \times \mathbf{0.5}))$

$+ -2(\mathbf{1.9} - (0 + 1 \times \mathbf{2.3}))$

$+ -2(\mathbf{3.2} - (0 + 1 \times \mathbf{2.9})) = \mathbf{-1.6}$

$\mathbf{Step\ Size}_{Intercept} = \mathbf{-1.6} \times \mathbf{Learning\ Rate}$

...now we plug the **Slopes** into the **Step Size** formulas...

$\frac{d}{d\,slope}$ Sum of squared residuals =

$-2 \times \mathbf{0.5}(\mathbf{1.4} - (0 + 1 \times \mathbf{0.5}))$

$+ -2 \times \mathbf{2.9}(\mathbf{3.2} - (0 + 1 \times \mathbf{2.9}))$

$+ -2 \times \mathbf{2.3}(\mathbf{1.9} - (0 + 1 \times \mathbf{2.3})) = \mathbf{-0.8}$

$\mathbf{Step\ Size}_{Slope} = \mathbf{-0.8} \times \mathbf{Learning\ Rate}$



$\frac{d}{d\,intercept}$ Sum of squared residuals =

$-2(\mathbf{1.4} - (0 + 1 \times \mathbf{0.5}))$

$+ -2(\mathbf{1.9} - (0 + 1 \times \mathbf{2.3}))$

$+ -2(\mathbf{3.2} - (0 + 1 \times \mathbf{2.9})) = \mathbf{-1.6}$

$\mathbf{Step\ Size}_{Intercept} = \mathbf{-1.6} \times \mathbf{Learning\ Rate}$

...and multiply by the **Learning Rate**, which this time we set to **0.01**...

$\frac{d}{d\,slope}$ Sum of squared residuals =

$-2 \times \mathbf{0.5}(\mathbf{1.4} - (0 + 1 \times \mathbf{0.5}))$

$+ -2 \times \mathbf{2.9}(\mathbf{3.2} - (0 + 1 \times \mathbf{2.9}))$

$+ -2 \times \mathbf{2.3}(\mathbf{1.9} - (0 + 1 \times \mathbf{2.3})) = \mathbf{-0.8}$

$\mathbf{Step\ Size}_{Slope} = \mathbf{-0.8} \times \mathbf{Learning\ Rate}$

$\frac{d}{d\,intercept}$ Sum of squared residuals =

$-2(1.4 - (0 + 1 \times 0.5)$
$+ -2(1.9 - (0 + 1 \times 2.3))$
$+ -2(3.2 - (0 + 1 \times 2.9)) = -1.6$

**Step Size**$_{Intercept}$ = -1.6 × 0.01

**NOTE:** The larger **Learning Rate** that we used in the first example doesn't work this time. Even after a bunch of steps, **Gradient Descent** doesn't arrive at the correct answer.

$\frac{d}{d\,slope}$ Sum of squared residuals =

$-2 \times 0.5(1.4 - (0 + 1 \times 0.5))$
$+ -2 \times 2.9(3.2 - (0 + 1 \times 2.9))$
$+ -2 \times 2.3(1.9 - (0 + 1 \times 2.3)) = -0.8$

**Step Size**$_{Slope}$ = -0.8 × 0.01



$\frac{d}{d\,intercept}$ Sum of squared residuals =

$-2(1.4 - (0 + 1 \times 0.5)$
$+ -2(1.9 - (0 + 1 \times 2.3))$
$+ -2(3.2 - (0 + 1 \times 2.9)) = -1.6$

**Step Size**$_{Intercept}$ = -1.6 × 0.01

This means that **Gradient Descent** can be very sensitive to the **Learning Rate**.

$\frac{d}{d\,slope}$ Sum of squared residuals =

$-2 \times 0.5(1.4 - (0 + 1 \times 0.5))$
$+ -2 \times 2.9(3.2 - (0 + 1 \times 2.9))$
$+ -2 \times 2.3(1.9 - (0 + 1 \times 2.3)) = -0.8$

**Step Size**$_{Slope}$ = -0.8 × 0.01

$\frac{d}{d\,intercept}$ Sum of squared residuals =

$-2(1.4 - (0 + 1 \times 0.5)$
$+ -2(1.9 - (0 + 1 \times 2.3))$
$+ -2(3.2 - (0 + 1 \times 2.9)) = -1.6$

**Step Size**$_{Intercept}$ = -1.6 × 0.01

The good news is that in practice, a reasonable **Learning Rate** can be determined automatically by starting large and getting smaller with each step.

$\frac{d}{d\,slope}$ Sum of squared residuals =

$-2 \times 0.5(1.4 - (0 + 1 \times 0.5))$
$+ -2 \times 2.9(3.2 - (0 + 1 \times 2.9))$
$+ -2 \times 2.3(1.9 - (0 + 1 \times 2.3)) = -0.8$

**Step Size**$_{Slope}$ = -0.8 × 0.01

$\frac{d}{d\,intercept}$ Sum of squared residuals =

$-2(1.4 - (0 + 1 \times 0.5)$
$+ -2(1.9 - (0 + 1 \times 2.3))$
$+ -2(3.2 - (0 + 1 \times 2.9)) = -1.6$

**Step Size**$_{Intercept}$ = -1.6 × 0.01

So, in theory, you shouldn't have to worry too much about the **Learning Rate**.

$\frac{d}{d\,slope}$ Sum of squared residuals =

$-2 \times 0.5(1.4 - (0 + 1 \times 0.5))$
$+ -2 \times 2.9(3.2 - (0 + 1 \times 2.9))$
$+ -2 \times 2.3(1.9 - (0 + 1 \times 2.3)) = -0.8$

**Step Size**$_{Slope}$ = -0.8 × 0.01

$\frac{d}{d\,intercept}$ Sum of squared residuals =

$-2(1.4 - (0 + 1 \times 0.5)$
$+ -2(1.9 - (0 + 1 \times 2.3))$
$+ -2(3.2 - (0 + 1 \times 2.9)) = -1.6$

**Step Size**$_{Intercept}$ = -1.6 × 0.01 = **-0.016**

Anyway, we do the math and get two **Step Sizes**.

$\frac{d}{d\,slope}$ Sum of squared residuals =

$-2 \times 0.5(1.4 - (0 + 1 \times 0.5))$
$+ -2 \times 2.9(3.2 - (0 + 1 \times 2.9))$
$+ -2 \times 2.3(1.9 - (0 + 1 \times 2.3)) = -0.8$

**Step Size**$_{Slope}$ = -0.8 × 0.01 = **-0.008**

$\frac{d}{d\,intercept}$ Sum of squared residuals =

$-2(1.4 - (0 + 1 \times 0.5)$
$+ -2(1.9 - (0 + 1 \times 2.3))$
$+ -2(3.2 - (0 + 1 \times 2.9)) = -1.6$

**Step Size**$_{Intercept}$ = -1.6 × 0.01 = **-0.016**

**New Intercept** = **Old Intercept** - **Step Size**

Now we calculate the **New Intercept** and **New Slope** by plugging in the **Old Intercept** and the **Old Slope**...

$\frac{d}{d\,slope}$ Sum of squared residuals =

$-2 \times 0.5(1.4 - (0 + 1 \times 0.5))$
$+ -2 \times 2.9(3.2 - (0 + 1 \times 2.9))$
$+ -2 \times 2.3(1.9 - (0 + 1 \times 2.3)) = -0.8$

**Step Size**$_{Slope}$ = -0.8 × 0.01 = **-0.008**

**New Slope** = **Old Slope** - **Step Size**

$\frac{d}{d\,intercept}$ Sum of squared residuals =

$-2(1.4 - (0 + 1 \times 0.5))$

$+ -2(1.9 - (0 + 1 \times 2.3))$

$+ -2(3.2 - (0 + 1 \times 2.9)) = -1.6$

**Step Size**$_{Intercept}$ = -1.6 × 0.01 = **-0.016**

**New Intercept** = 0 - (-0.016)

…and the **Step Sizes**…

$\frac{d}{d\,slope}$ Sum of squared residuals =

$-2 \times 0.5(1.4 - (0 + 1 \times 0.5))$

$+ -2 \times 2.9(3.2 - (0 + 1 \times 2.9))$

$+ -2 \times 2.3(1.9 - (0 + 1 \times 2.3)) = -0.8$

**Step Size**$_{Slope}$ = -0.8 × 0.01 = **-0.008**

**New Slope** = 1 - (-0.008)



$\frac{d}{d\,intercept}$ Sum of squared residuals =

$-2(1.4 - (0 + 1 \times 0.5))$

$+ -2(1.9 - (0 + 1 \times 2.3))$

$+ -2(3.2 - (0 + 1 \times 2.9)) = -1.6$

**Step Size**$_{Intercept}$ = -1.6 × 0.01 = **-0.016**

**New Intercept** = 0 - (-0.016) = 0.016

…and we end up with a **New Intercept** and a **New Slope**.

$\frac{d}{d\,slope}$ Sum of squared residuals =

$-2 \times 0.5(1.4 - (0 + 1 \times 0.5))$

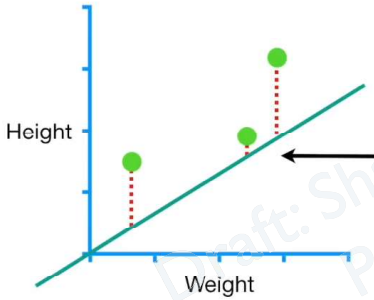
$+ -2 \times 2.9(3.2 - (0 + 1 \times 2.9))$

$+ -2 \times 2.3(1.9 - (0 + 1 \times 2.3)) = -0.8$

**Step Size**$_{Slope}$ = -0.8 × 0.01 = **-0.008**

**New Slope** = 1 - (-0.008) = 1.008



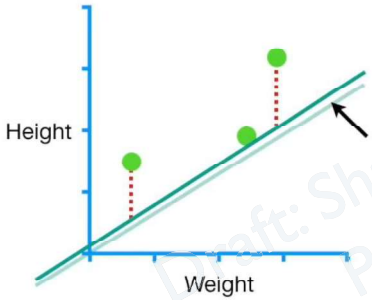



**New Intercept** = 0 - (-0.016) = 0.016

This is the line we started with… (**Slope = 1** and **Intercept = 0**)

**New Slope** = 1 - (-0.008) = 1.008



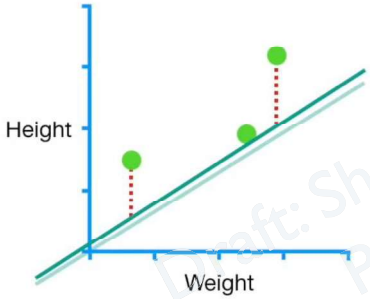



**New Intercept** = 0 - (-0.016) = 0.016

…and this is the new line (with **Slope = 1.008** and **Intercept = 0.016**) after the first step.

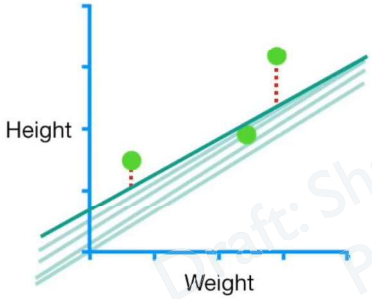
**New Slope** = 1 - (-0.008) = 1.008



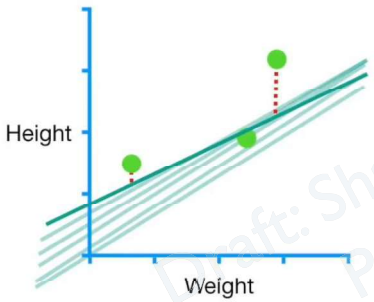



Now we just repeat what we did until all of the **Steps Sizes** are very small or we reach the **Maximum Number of Steps**.

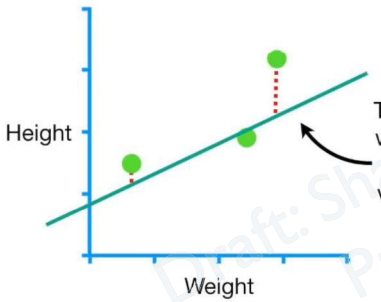




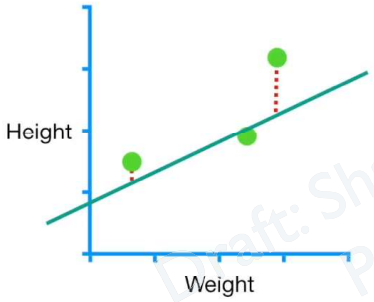

Now we just repeat what we did until all of the **Steps Sizes** are very small or we reach the **Maximum Number of Steps**.



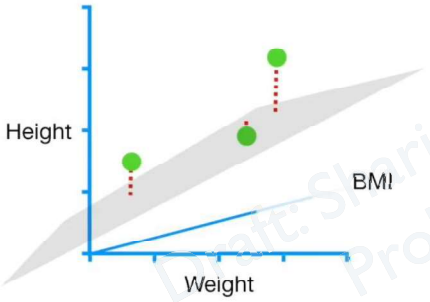

Now we just repeat what we did until all of the **Steps Sizes** are very small or we reach the **Maximum Number of Steps**.




This is the best fitting line, with **Intercept** = **0.95** and **Slope** = **0.64**, the same values we get from **Least Squares**.



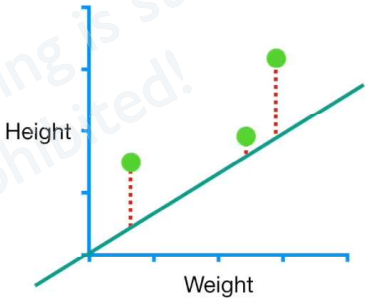

We now know how **Gradient Descent** optimizes two parameters, the **Slope** and **Intercept**.




If we had more parameters, then we'd just take more derivatives and everything else stays the same.

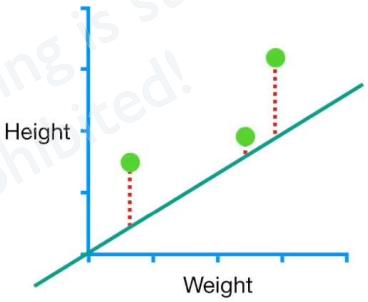
Sum of squared residuals = (**1.4** - (intercept + 0.64 × **0.5**))$^2$
+ (**1.9** - (intercept + 0.64 × **2.3**))$^2$
+ (**3.2** - (intercept + 0.64 × **2.9**))$^2$

**NOTE:** The Sum of the Squared Residuals is just one type of **Loss Function**.




Sum of squared residuals = (**1.4** - (intercept + 0.64 × **0.5**))$^2$
+ (**1.9** - (intercept + 0.64 × **2.3**))$^2$
+ (**3.2** - (intercept + 0.64 × **2.9**))$^2$

However, there are tons of other **Loss Functions** that work with other types of data.



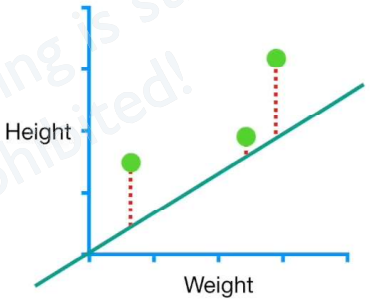
Sum of squared residuals = (**1.4** - (intercept + 0.64 × **0.5**))$^2$

+ (**1.9** - (intercept + 0.64 × **2.3**))$^2$

+ (**3.2** - (intercept + 0.64 × **2.9**))$^2$

However, there are tons of other **Loss Functions** that work with other types of data.

Regardless of which **Loss Function** you use, **Gradient Descent** works the same way.




**Step 1:** Take the derivative of the **Loss Function** for each parameter in it.

**Step 1:** Take the derivative of the **Loss Function** for each parameter in it.
In fancy Machine Learning Lingo, take the **Gradient** of the **Loss Function**.

**Step 1:** Take the derivative of the **Loss Function** for each parameter in it.
In fancy Machine Learning Lingo, take the **Gradient** of the **Loss Function**.

**Step 2:** Pick random values for the parameters.

**Step 1:** Take the derivative of the **Loss Function** for each parameter in it.
In fancy Machine Learning Lingo, take the **Gradient** of the **Loss Function**.

**Step 2:** Pick random values for the parameters.

**Step 3:** Plug the parameter values into the derivatives (ahem, the **Gradient**).

**Step 1:** Take the derivative of the **Loss Function** for each parameter in it.
In fancy Machine Learning Lingo, take the **Gradient** of the **Loss Function**.

**Step 2:** Pick random values for the parameters.

**Step 3:** Plug the parameter values into the derivatives (ahem, the **Gradient**).

**Step 4:** Calculate the Step Sizes: **Step Size** = **Slope** × **Learning Rate**

**Step 1:** Take the derivative of the **Loss Function** for each parameter in it. In fancy Machine Learning Lingo, take the **Gradient** of the **Loss Function**.

**Step 2:** Pick random values for the parameters.

**Step 3:** Plug the parameter values into the derivatives (ahem, the **Gradient**).

**Step 4:** Calculate the Step Sizes: **Step Size** = **Slope** × **Learning Rate**

**Step 5:** Calculate the New Parameters:

**New Parameter** = **Old Parameter** - **Step Size**

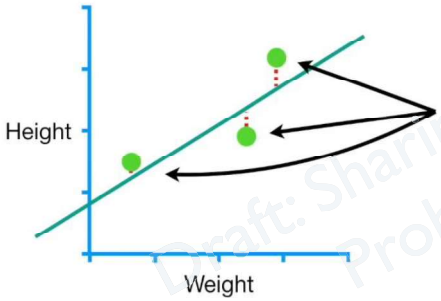
Now go back to **Step 3** and repeat until **Step Size** is very small, or you reach the **Maximum Number of Steps**.

**Step 3:** Plug the parameter values into the derivatives (ahem, the **Gradient**).

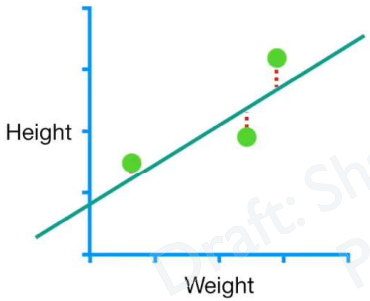
**Step 4:** Calculate the Step Sizes: **Step Size** = **Slope** × **Learning Rate**

**Step 5:** Calculate the New Parameters:

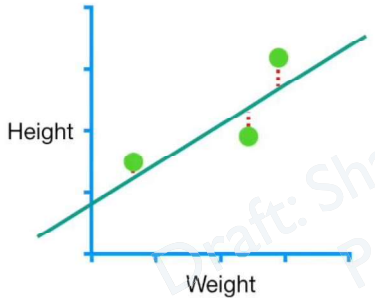
**New Parameter** = **Old Parameter** - **Step Size**



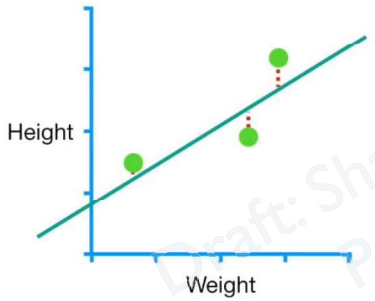

In our example, we only had three data points, so the math didn't take very long…




So there is a thing called **Stochastic Gradient Descent** that uses a randomly selected subset of the data at every step rather than the full dataset.

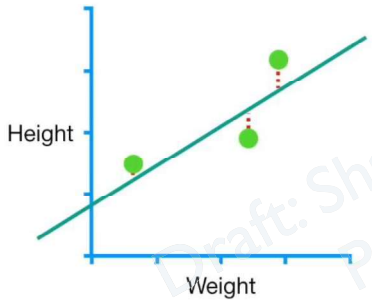



…but when you have millions of data points, it can take a long time.




So there is a thing called **Stochastic Gradient Descent** that uses a randomly selected subset of the data at every step rather than the full dataset.

This reduces the time spent calculating the derivatives of the **Loss Function**.




So there is a thing called **Stochastic Gradient Descent** that uses a randomly selected subset of the data at every step rather than the full dataset.

This reduces the time spent calculating the derivatives of the **Loss Function**.

That's all.

**Stochastic Gradient Descent** sounds fancy, but it's no big deal.

But what if we had a more complicated model, like a **Logistic Regression** that used **23,000** genes to predict if someone will have a disease?

$\frac{d}{d\,gene1}$ Loss Function()

$\frac{d}{d\,gene2}$ Loss Function()

$\frac{d}{d\,gene3}$ Loss Function()

$\frac{d}{d\,gene4}$ Loss Function()

$\frac{d}{d\,gene5}$ Loss Function()

$\frac{d}{d\,gene6}$ Loss Function()

$\frac{d}{d\,gene7}$ Loss Function()

etc…etc…etc…

Then we would have **23,000** derivatives to plug the data into.

$\frac{d}{d\,gene1}$ Loss Function()

$\frac{d}{d\,gene2}$ Loss Function()

$\frac{d}{d\,gene3}$ Loss Function()

$\frac{d}{d\,gene4}$ Loss Function()

$\frac{d}{d\,gene5}$ Loss Function()

$\frac{d}{d\,gene6}$ Loss Function()

$\frac{d}{d\,gene7}$ Loss Function()

etc…etc…etc…

And what if we had data from **1,000,000** samples?

$\frac{d}{d\,gene1}$ Loss Function()

$\frac{d}{d\,gene2}$ Loss Function()

$\frac{d}{d\,gene3}$ Loss Function()

$\frac{d}{d\,gene4}$ Loss Function()

$\frac{d}{d\,gene5}$ Loss Function()

$\frac{d}{d\,gene6}$ Loss Function()

$\frac{d}{d\,gene7}$ Loss Function()

etc…etc…etc…

Then we would have to calculate **1,000,000** terms for each of the **23,000** derivatives.

$\frac{d}{d\,gene1}$ Loss Function()

$\frac{d}{d\,gene2}$ Loss Function()

$\frac{d}{d\,gene3}$ Loss Function()

$\frac{d}{d\,gene4}$ Loss Function()

$\frac{d}{d\,gene5}$ Loss Function()

$\frac{d}{d\,gene6}$ Loss Function()

$\frac{d}{d\,gene7}$ Loss Function()

etc…etc…etc…

Then we would have to calculate **1,000,000** terms for each of the **23,000** derivatives.

In other words, we'd have to calculate **23,000,000,000** terms for each step.

$\frac{d}{d\,gene1}$ Loss Function()

$\frac{d}{d\,gene2}$ Loss Function()

$\frac{d}{d\,gene3}$ Loss Function()

$\frac{d}{d\,gene4}$ Loss Function()

$\frac{d}{d\,gene5}$ Loss Function()

$\frac{d}{d\,gene6}$ Loss Function()

$\frac{d}{d\,gene7}$ Loss Function()

etc…etc…etc…

Then we would have to calculate **1,000,000** terms for each of the **23,000** derivatives.

In other words, we'd have to calculate **23,000,000,000** terms for each step.

And since it is common to take at least **1,000** steps, we would calculate at least **2,300,000,000,000** terms.

$\frac{d}{d\,gene1}$ Loss Function()

$\frac{d}{d\,gene2}$ Loss Function()

$\frac{d}{d\,gene3}$ Loss Function()

$\frac{d}{d\,gene4}$ Loss Function()

$\frac{d}{d\,gene5}$ Loss Function()

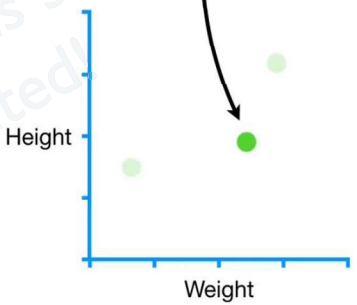
$\frac{d}{d\,gene6}$ Loss Function()

$\frac{d}{d\,gene7}$ Loss Function()

etc…etc…etc…

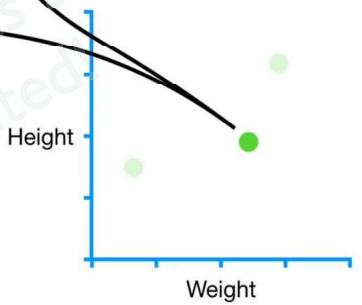
This is where **Stochastic Gradient Descent** comes in handy.

**Stochastic Gradient Descent** would randomly pick one sample for each step…




$\frac{d}{d\,intercept}$ Sum of squared residuals =
-2(**Height** - (intercept + slope × **Weight**)

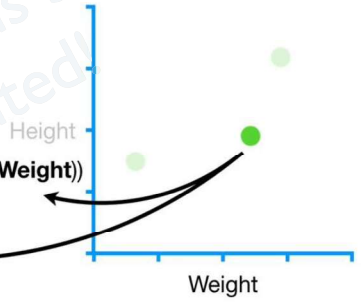
…and just use that one sample to calculate the derivatives.




$\frac{d}{d\,intercept}$ Sum of squared residuals =
-2(**Height** - (intercept + slope × **Weight**)

…and just use that one sample to calculate the derivatives.

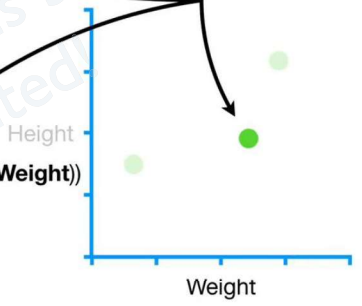
$\frac{d}{d\,slope}$ Sum of squared residuals =
-2 × **Weight**(**Height** - (intercept + slope × **Weight**))




$\frac{d}{d\,intercept}$ Sum of squared residuals =
-2(**Height** - (intercept + slope × **Weight**)

Thus, in this super simple example, **Stochastic Gradient Descent** reduced the number of terms computed by a factor of **3**.

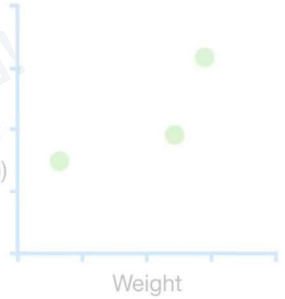
$\frac{d}{d\,slope}$ Sum of squared residuals =
-2 × **Weight**(**Height** - (intercept + slope × **Weight**))



$\frac{d}{d\ intercept}$ Sum of squared residuals =
-2(**Height** - (intercept + slope × **Weight**)

$\frac{d}{d\ slope}$ Sum of squared residuals =
-2 × **Weight**(**Height** - (intercept + slope × **Weight**))

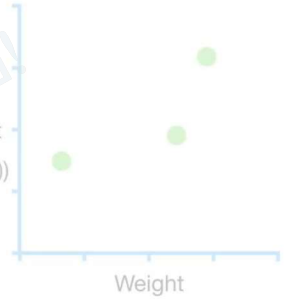
If we had **1,000,000** samples, then **Stochastic Gradient Descent** would reduce the amount terms computed by a factor of **1,000,000**.



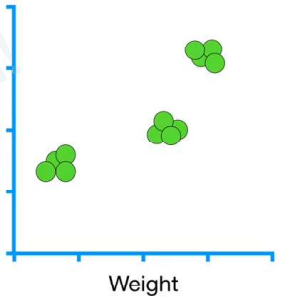

$\frac{d}{d\ intercept}$ Sum of squared residuals =
-2(**Height** - (intercept + slope × **Weight**)

$\frac{d}{d\ slope}$ Sum of squared residuals =
-2 × **Weight**(**Height** - (intercept + slope × **Weight**))

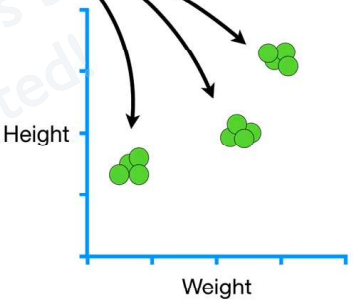
So that's pretty cool.



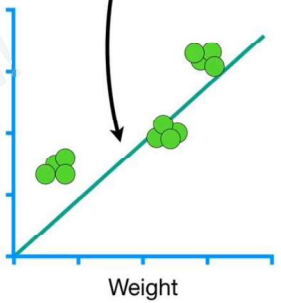

**Stochastic Gradient Descent** is especially useful when there are redundancies in the data.



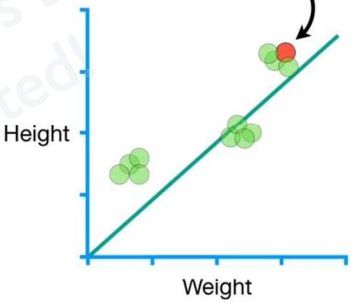

For example, we have **12** data points, but there is a lot of redundancy that forms **3** clusters.




So we start with a line with the **intercept** = **0** and the **slope** = **1**…



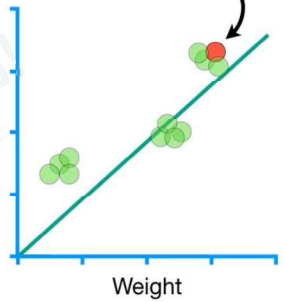

…then we randomly pick this point…



$\frac{d}{d\ intercept}$ Sum of squared residuals = -2(**Height** - (0 + 1 × **Weight**)

…so we plug in the **Weight**, **3**…

$\frac{d}{d\ slope}$ Sum of squared residuals = -2 × **Weight**(**Height** - (0 + 1 × **Weight**))

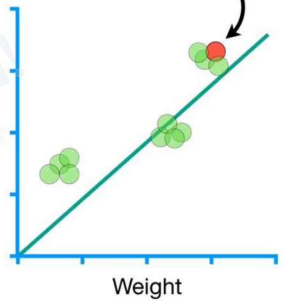





$\frac{d}{d\ intercept}$ Sum of squared residuals = -2(**3.3** - (0 + 1 × **3**))

…and **Height**, **3.3**…

$\frac{d}{d\ slope}$ Sum of squared residuals = -2 × **3**(**3.3** - (0 + 1 × **3**))





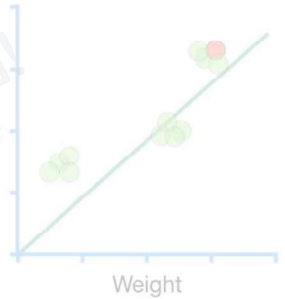

$\frac{d}{d\ intercept}$ Sum of squared residuals = -2(**3.3** - (0 + 1 × **3**)) = -0.6

**Step Size**$_{Intercept}$ = **Slope** × **Learning Rate**

…plug in the slopes…

**Step Size**$_{Slope}$ = **Slope** × **Learning Rate**

$\frac{d}{d\ slope}$ Sum of squared residuals = -2 × **3**(**3.3** - (0 + 1 × **3**)) = -1.8






$\frac{d}{d\ intercept}$ Sum of squared residuals = -2(**3.3** - (0 + 1 × **3**)) = -0.6

…then multiply by the **Learning Rate**.

**Step Size**$_{Intercept}$ = -0.6 × **Learning Rate**

**Step Size**$_{Slope}$ = -1.8 × **Learning Rate**

$\frac{d}{d\ slope}$ Sum of squared residuals = -2 × **3**(**3.3** - (0 + 1 × **3**)) = -1.8



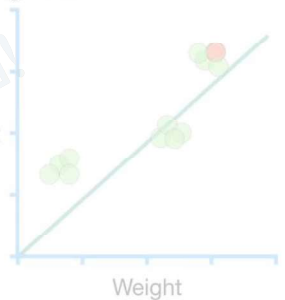

$\frac{d}{d\ intercept}$ Sum of squared residuals = -2(**3.3** - (0 + 1 × **3**)) = -0.6

**Step Size**$_{Intercept}$ = -0.6 × **Learning Rate**

**Step Size**$_{Slope}$ = -1.8 × **Learning Rate**

**NOTE:** Just like with regular **Gradient Descent**, **Stochastic Gradient Descent** is sensitive to the value you choose for the **Learning Rate**…

$\frac{d}{d\ slope}$ Sum of squared residuals = -2 × **3**(**3.3** - (0 + 1 × **3**)) = -1.8





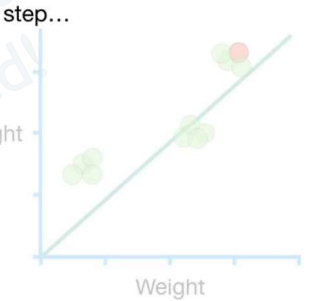

$\frac{d}{d\ intercept}$ Sum of squared residuals = -2(**3.3** - (0 + 1 × **3**)) = -0.6

**Step Size**$_{Intercept}$ = -0.6 × **Learning Rate**

**Step Size**$_{Slope}$ = -1.8 × **Learning Rate**

…and just like for regular **Gradient Descent**, the general strategy is to start with a relatively *large* **Learning Rate** and make it *smaller* with each step…

$\frac{d}{d\ slope}$ Sum of squared residuals = -2 × **3**(**3.3** - (0 + 1 × **3**)) = -1.8



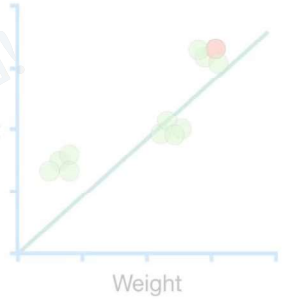
$\frac{d}{d\,intercept}$ Sum of squared residuals = -2(**3.3** - (0 + 1 × **3**) = -0.6

**Step Size**$_{Intercept}$ = -0.6 × **Learning Rate**

**Step Size**$_{Slope}$ = -1.8 × **Learning Rate**

$\frac{d}{d\,slope}$ Sum of squared residuals = -2 × **3**(**3.3** - (0 + 1 × **3**)) = -1.8

…and lastly, just like for regular **Gradient Descent**, many implementations of **Stochastic Gradient Descent** will take care of this for you by default.






$\frac{d}{d\,intercept}$ Sum of squared residuals = -2(**3.3** - (0 + 1 × **3**) = -0.6

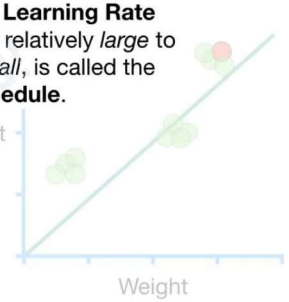
**Step Size**$_{Intercept}$ = -0.6 × **Learning Rate**

**Step Size**$_{Slope}$ = -1.8 × **Learning Rate**

**TERMINOLOGY ALERT!!!**

The way the **Learning Rate** changes, from relatively *large* to relatively *small*, is called the **schedule**.

$\frac{d}{d\,slope}$ Sum of squared residuals = -2 × **3**(**3.3** - (0 + 1 × **3**)) = -1.8




Draft: Sharing is strictly Prohibited!

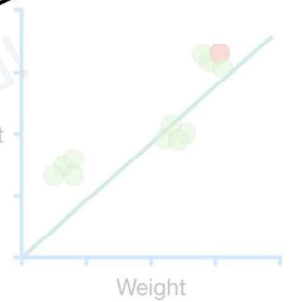
$\frac{d}{d\,intercept}$ Sum of squared residuals = -2(**3.3** - (0 + 1 × **3**) = -0.6

In this simple example, however, we'll just setting the **Learning Rate** to **0.01**.

**Step Size**$_{Intercept}$ = -0.6 × 0.01

**Step Size**$_{Slope}$ = -1.8 × 0.01

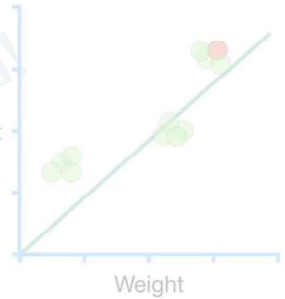
$\frac{d}{d\,slope}$ Sum of squared residuals = -2 × **3**(**3.3** - (0 + 1 × **3**)) = -1.8




Draft: Sharing is strictly Prohibited!

**New Intercept** = 0 - -0.006 = 0.006

**New Slope** = 1 - -0.018 = 1.018

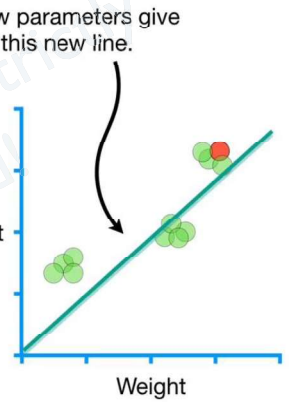



Draft: Sharing is strictly Prohibited!

**New Intercept** = 0 - -0.006 = 0.006

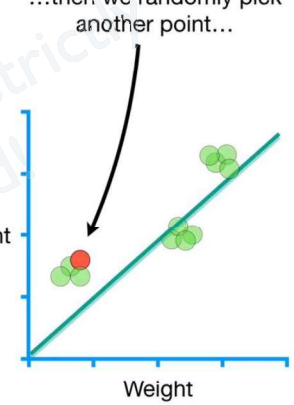
**New Slope** = 1 - -0.018 = 1.018

The new parameters give us this new line.

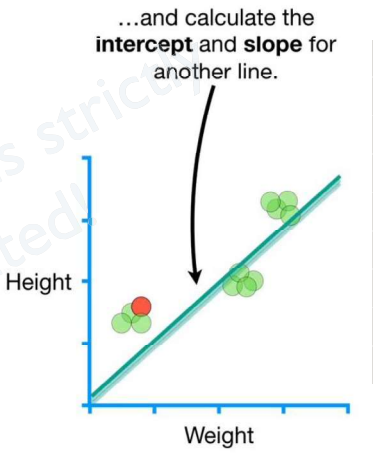


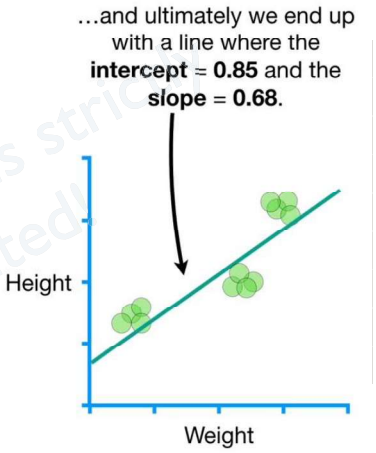

Draft: Sharing is strictly Prohibited!

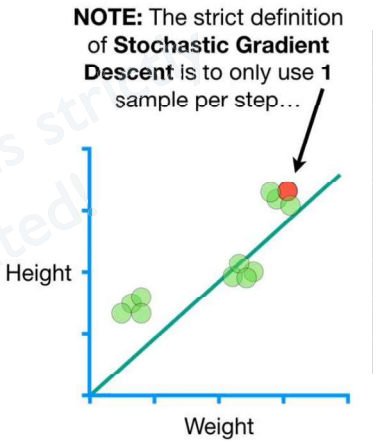
…then we randomly pick another point…

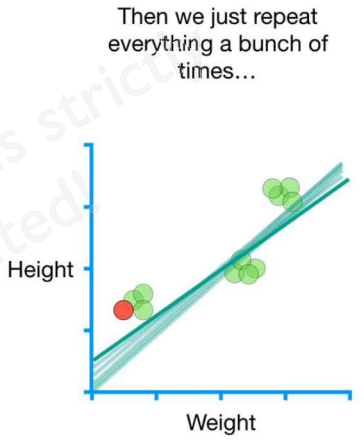



Draft: Sharing is strictly Prohibited!

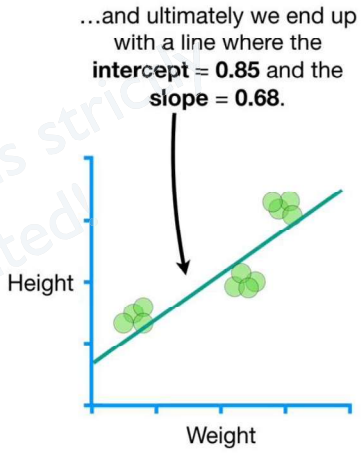


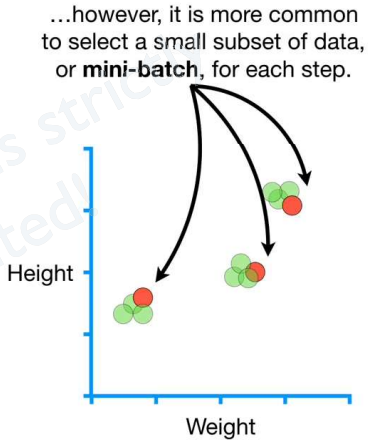



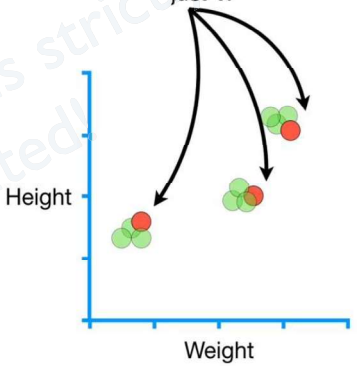




…and the least squares estimates, aka, the gold standard, gives a line where the **intercept** = **0.87** and the **slope** = **0.68**.

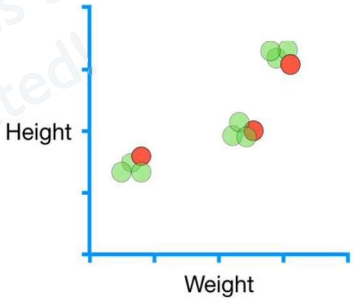


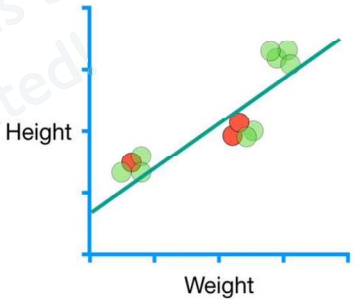
For example, we could use **3** samples per step, instead of just **1**.



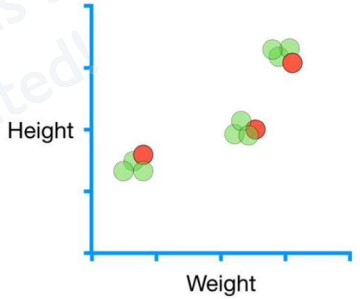

Similar to using all of the data, using a **mini-batch** can result in more stable estimates of the parameters in fewer steps…



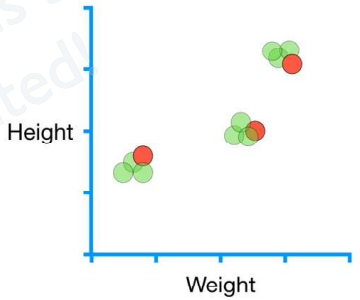

In this example, using **3** samples per step we ended up with the **intercept** = **0.86** and the **slope** = **0.68**.




Using a **mini-batch** for each step takes the best of both worlds between using just one sample and all of of the data at each step.

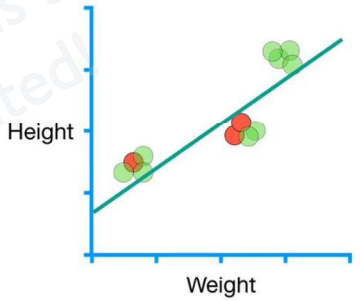



…and like using just one sample per step, using a **mini-batch** is much faster than using all of the data.

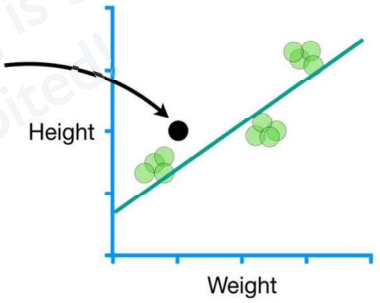



In this example, using **3** samples per step we ended up with the **intercept** = **0.86** and the **slope** = **0.68**.

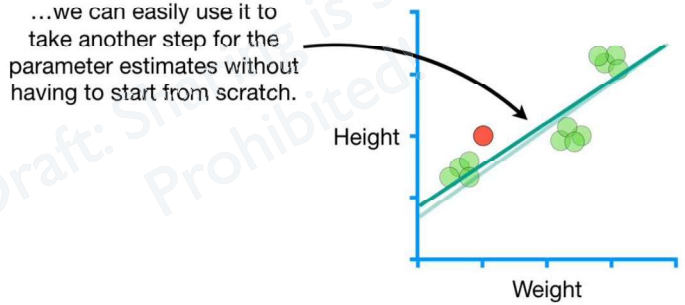
…which means that the estimate for the intercept was just a little closer to the gold standard, **0.87**, then when we used one sample and got **0.85**.



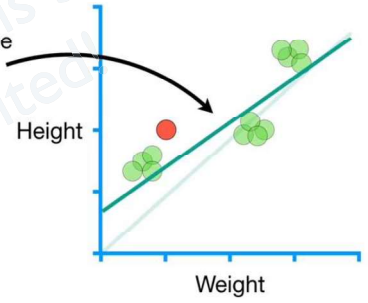
One cool thing about **Stochastic Gradient Descent** is that when we get new data…



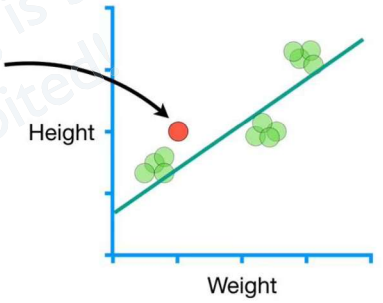

…we can easily use it to take another step for the parameter estimates without having to start from scratch.



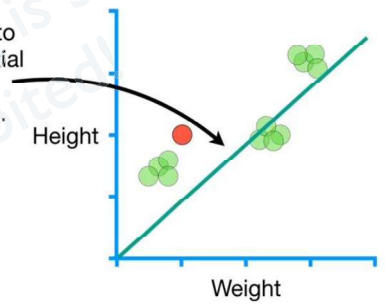

Instead, we pick up right where we left off and take one more step using the new sample.




…we can easily use it to take another step for the parameter estimates without having to start from scratch.

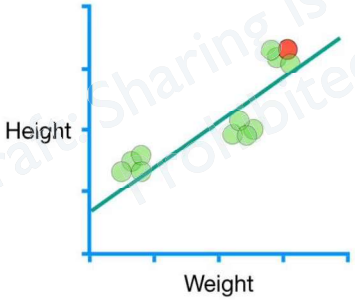



In other words, we don't have to go all of the way back to the initial guesses for the **slope** and **intercept** and redo everything.




# In Summary…

**Stochastic Gradient Descent** is just like regular **Gradient Descent**, except it only looks at one sample per step…




$\frac{d}{d\,gene1}$ Loss Function()

$\frac{d}{d\,gene2}$ Loss Function()

$\frac{d}{d\,gene3}$ Loss Function()

$\frac{d}{d\,gene4}$ Loss Function()

$\frac{d}{d\,gene5}$ Loss Function()

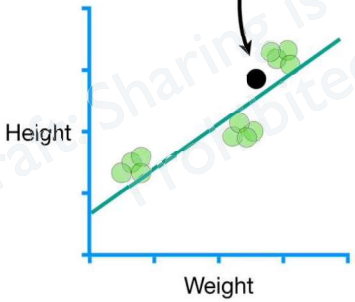
$\frac{d}{d\,gene6}$ Loss Function()

$\frac{d}{d\,gene7}$ Loss Function()

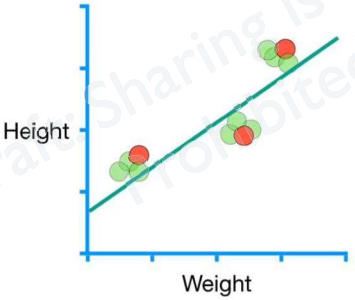
etc…etc…etc…

**Stochastic Gradient Descent** is great when we have tons of data and a lot of parameters.

And it's cool that we can easily update the parameters when new data shows up.




…or a small subset, or **mini-batch**, for each step.




$\frac{d}{d\,gene1}$ Loss Function()

$\frac{d}{d\,gene2}$ Loss Function()

$\frac{d}{d\,gene3}$ Loss Function()

$\frac{d}{d\,gene4}$ Loss Function()

$\frac{d}{d\,gene5}$ Loss Function()

$\frac{d}{d\,gene6}$ Loss Function()

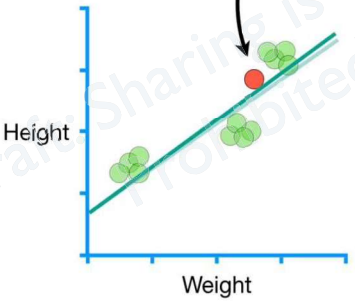
$\frac{d}{d\,gene7}$ Loss Function()

etc…etc…etc…

**Stochastic Gradient Descent** is great when we have tons of data and a lot of parameters.

In these situations, regular **Gradient Descent** may not be computationally feasible.

And it's cool that we can easily update the parameters when new data shows up.



# THANK YOU!