

Chapter 05

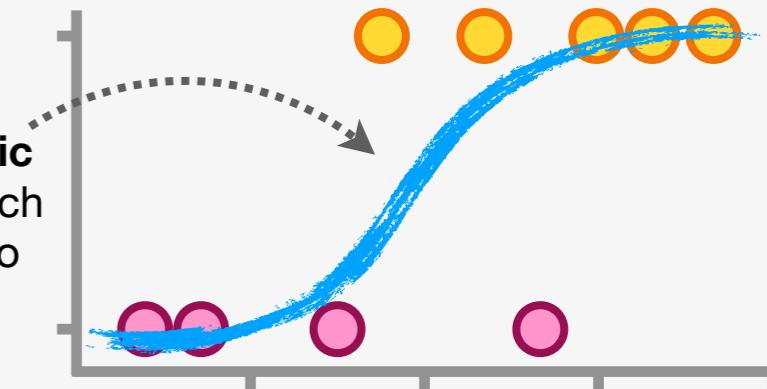
Gradient Descent!!!

Gradient Descent: Main Ideas

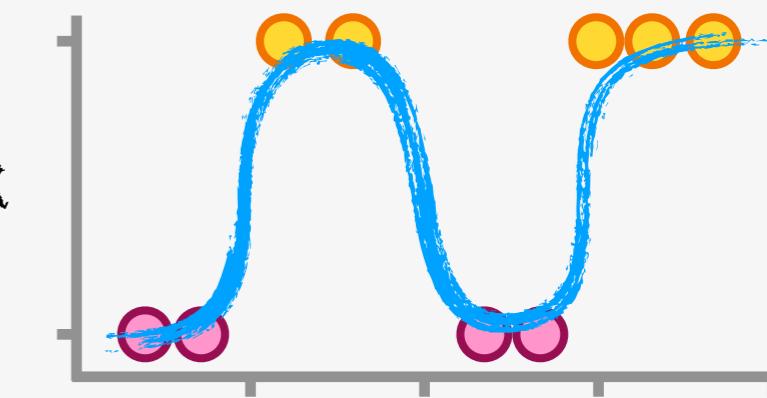
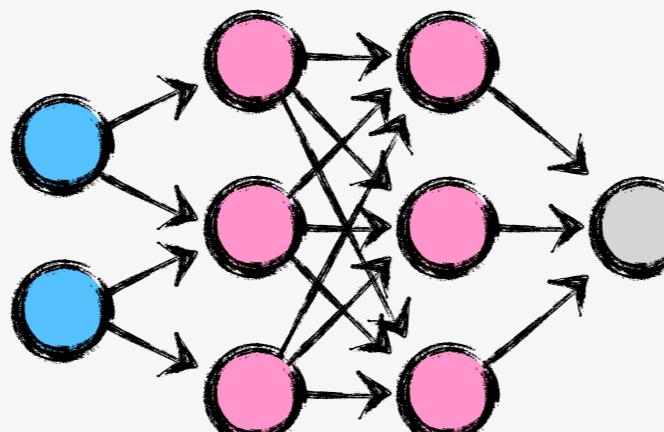
1

The Problem: A major part of machine learning is optimizing a model's fit to the data. Sometimes this can be done with an analytical solution, but it's not always possible.

For example, there is no analytical solution for **Logistic Regression (Chapter 6)**, which fits an **s-shaped squiggle** to data.



Likewise, there is no analytical solution for **Neural Networks (Chapter 12)**, which fit **fancy squiggles** to data.



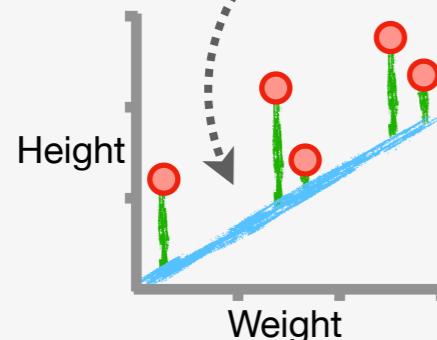
2

A Solution: When there's no analytical solution, **Gradient Descent** can save the day!

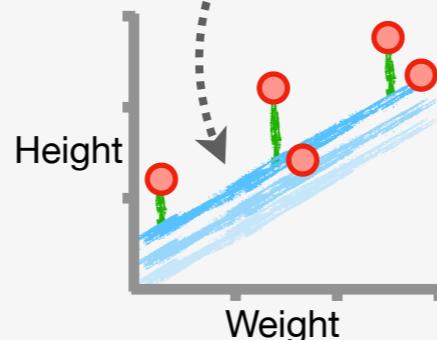
Gradient Descent is an *iterative solution* that incrementally steps toward an optimal solution and is used in a very wide variety of situations.

3

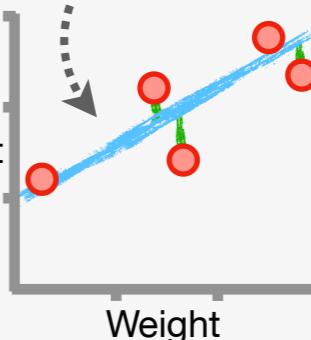
Gradient Descent starts with an initial guess...



...and then improves the guess, one step at a time...



...until it finds an optimal solution or reaches a maximum number of steps.

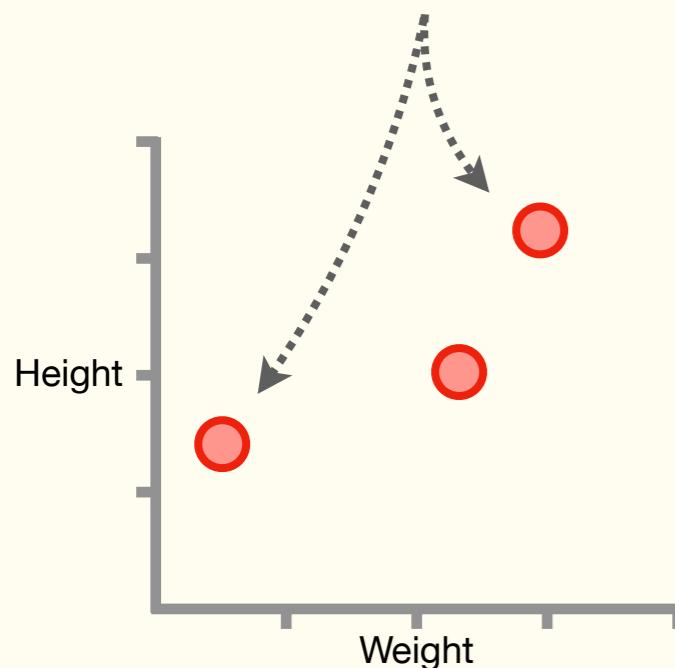


BAM!!!

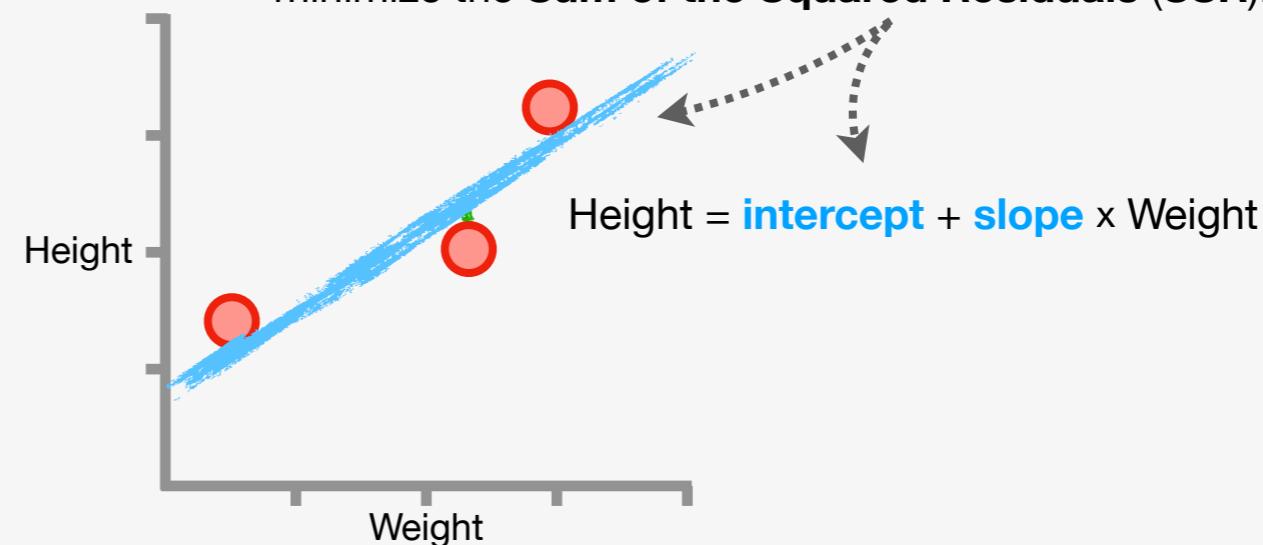
Gradient Descent: Details Part 1

NOTE: Even though there's an analytical solution for **Linear Regression**, we're using it to demonstrate how **Gradient Descent** works because we can compare the output from **Gradient Descent** to the known optimal values.

- Let's show how **Gradient Descent** fits a line to these Height and Weight measurements.



- Specifically, we'll show how **Gradient Descent** estimates the **intercept** and the **slope** of this line so that we minimize the **Sum of the Squared Residuals (SSR)**.

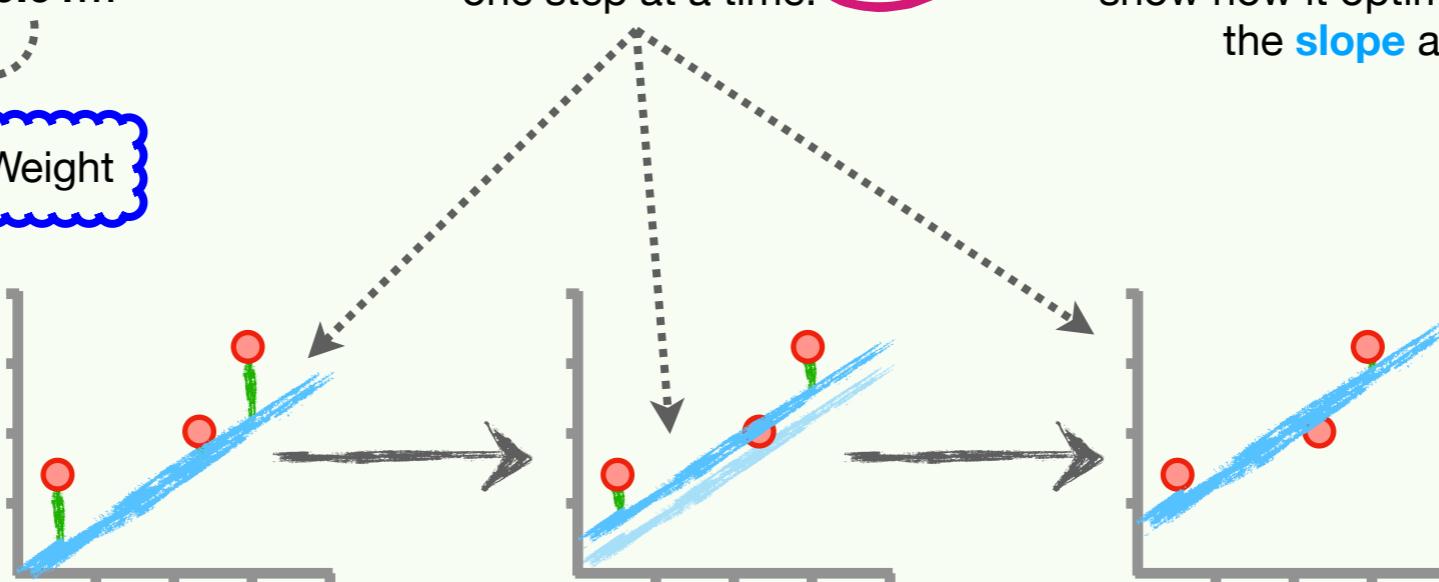


- To keep things simple at the start, let's plug in the analytical solution for the **slope**, **0.64**...

$$\text{Height} = \text{intercept} + 0.64 \times \text{Weight}$$

...and show how **Gradient Descent** optimizes the **intercept** one step at a time.

Once we understand how **Gradient Descent** optimizes the **intercept**, we'll show how it optimizes the **intercept** and the **slope** at the same time.



Gradient Descent: Details Part 2

4

In this example, we're fitting a line to data, and we can evaluate how well that line fits with the **Sum of the Squared Residuals (SSR)**.

Remember, **Residuals** are the difference between the Observed and Predicted values...

5

Now, because we have **3** data points, and thus, **3 Residuals**, the **SSR** has **3 terms**.

$$\begin{aligned} \text{SSR} = & (\text{Observed Height}_1 - (\text{intercept} + 0.64 \times \text{Weight}_1))^2 \\ & + (\text{Observed Height}_2 - (\text{intercept} + 0.64 \times \text{Weight}_2))^2 \\ & + (\text{Observed Height}_3 - (\text{intercept} + 0.64 \times \text{Weight}_3))^2 \end{aligned}$$

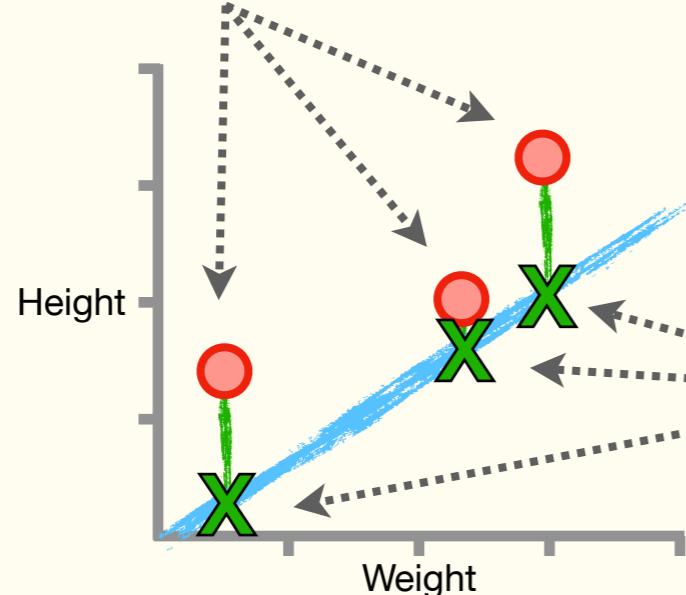
Residual = (Observed Height - Predicted Height) = (Height - (**intercept** + 0.64 × Weight))

...and the Observed Heights are the values we originally measured...

...and the Predicted Heights come from the equation for the line...

...so we can plug the equation for the line in for the Predicted value.

Psst! Don't forget to read **Step 5!!!**



$$\text{Predicted Height} = \text{intercept} + 0.64 \times \text{Weight}$$

Gradient Descent: Details Part 3

6

In this first example, since we're only optimizing the y-axis **intercept**, we'll start by assigning it a random value. In this case, we'll initialize the **intercept** by setting it to **0**.

$$\text{Height} = 0 + 0.64 \times \text{Weight}$$

7

Now, to calculate the **SSR**, we first plug the value for the y-axis **intercept**, **0**, into the equation we derived in **Steps 4 and 5**...

$$\begin{aligned} \text{SSR} &= (\text{Observed Height}_1 - (\text{intercept} + 0.64 \times \text{Weight}_1))^2 \\ &\quad + (\text{Observed Height}_2 - (\text{intercept} + 0.64 \times \text{Weight}_2))^2 \\ &\quad + (\text{Observed Height}_3 - (\text{intercept} + 0.64 \times \text{Weight}_3))^2 \end{aligned}$$

$$\begin{aligned} \text{SSR} &= (\text{Observed Height}_1 - (0 + 0.64 \times \text{Weight}_1))^2 \\ &\quad + (\text{Observed Height}_2 - (0 + 0.64 \times \text{Weight}_2))^2 \\ &\quad + (\text{Observed Height}_3 - (0 + 0.64 \times \text{Weight}_3))^2 \end{aligned}$$

8

...then we plug in the **Observed** values for Height and Weight for each data point.

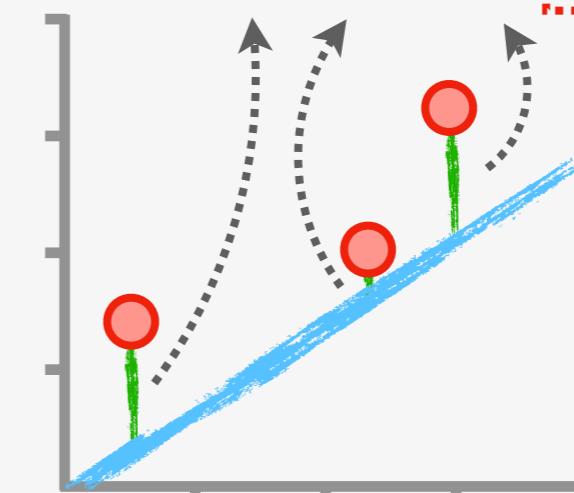
$$\begin{aligned} \text{SSR} &= (\text{Observed Height}_1 - (0 + 0.64 \times \text{Weight}_1))^2 \\ &\quad + (\text{Observed Height}_2 - (0 + 0.64 \times \text{Weight}_2))^2 \\ &\quad + (\text{Observed Height}_3 - (0 + 0.64 \times \text{Weight}_3))^2 \end{aligned}$$

$$\begin{aligned} \text{SSR} &= (1.4 - (0 + 0.64 \times 0.5))^2 \\ &\quad + (1.9 - (0 + 0.64 \times 2.3))^2 \\ &\quad + (3.2 - (0 + 0.64 \times 2.9))^2 \end{aligned}$$

9

Lastly, we just do the math. The **SSR** for when the y-axis **intercept** is set to **0** is **3.1**. Bam!

$$1.1^2 + 0.4^2 + 1.3^2 = 3.1$$

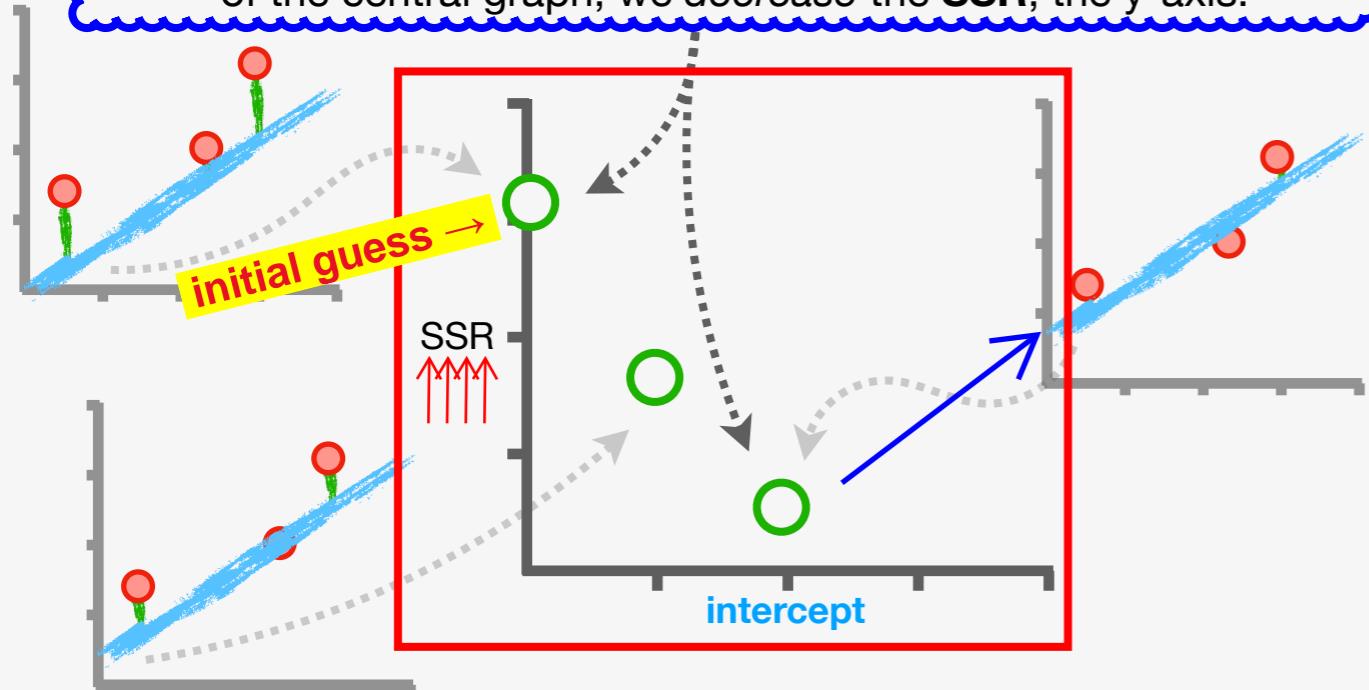


Gradient Descent: Details Part 1

Feature Scaling: this is done to reach the global min quickly by smoothening the curve → here we are ↓ the distance b/w parameters
Eg: Unit Scaling: 0-255 is converted to value b/w 0-1
Use: Linear Regressing, K-means or KNN (Euclidean Distance used)
Types: Normalization & Standardization

10

Now, because the goal is to minimize the **SSR**, it's a type of **Loss** or **Cost Function** (see **Terminology Alert** on the right). In **Gradient Descent**, we minimize the **Loss** or **Cost Function** by taking steps away from the initial guess toward the optimal value. In this case, we see that as we *increase* the **intercept**, the x-axis of the central graph, we *decrease* the **SSR**, the y-axis.



TERMINOLOGY ALERT!!!

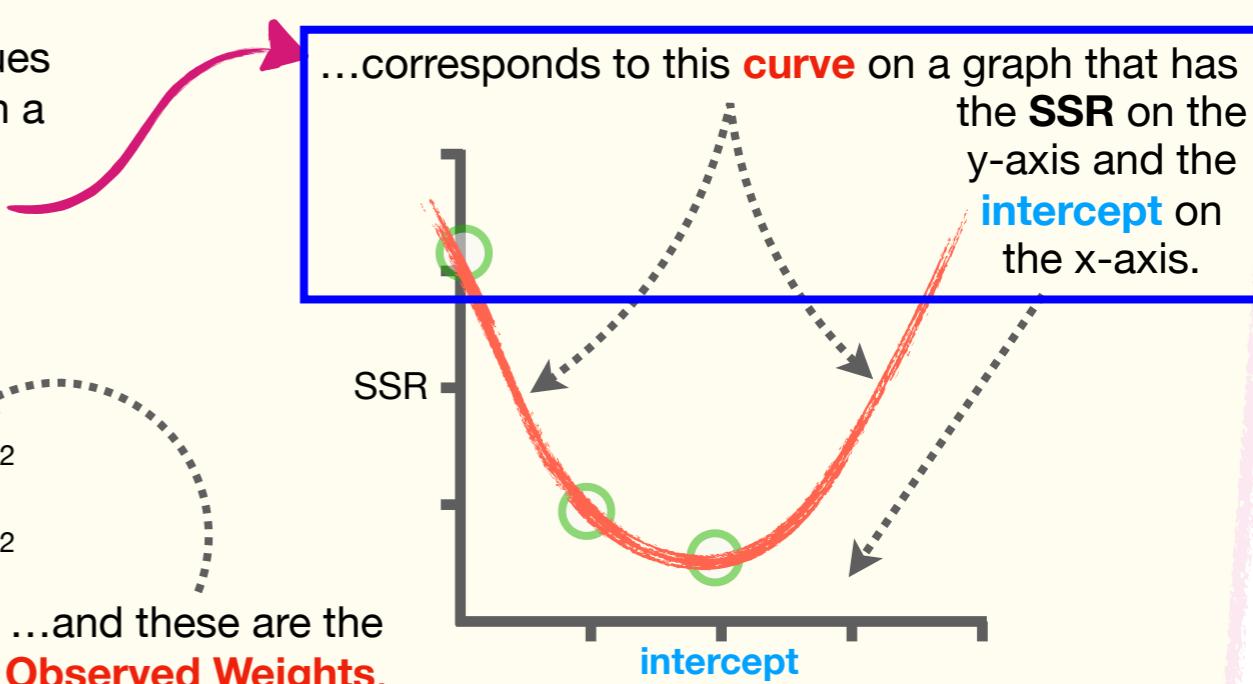
The terms **Loss Function** and **Cost Function** refer to anything we want to optimize when we fit a model to data. For example, we want to optimize the **SSR** or the **Mean Squared Error (MSE)** when we fit a straight line with **Regression** or a squiggly line (in **Neural Networks**). That said, some people use the term **Loss Function** to specifically refer to a function (like the **SSR**) applied to *only one data point*, and use the term **Cost Function** to specifically refer to a function (like the **SSR**) applied to *all* of the data. Unfortunately, these specific meanings are not universal, so be aware of the context and be prepared to be flexible. In this book, we'll use them together and interchangeably, as in “The **Loss** or **Cost Function** is the **SSR**. ”

11

However, rather than just randomly trying a bunch of values for the y-axis **intercept** and plotting the resulting **SSR** on a graph, we can plot the **SSR** as a function of the y-axis **intercept**. In other words, this equation for the **SSR**...

$$\text{SSR} = (1.4 - (\text{intercept} + 0.64 \times 0.5))^2 + (1.9 - (\text{intercept} + 0.64 \times 2.3))^2 + (3.2 - (\text{intercept} + 0.64 \times 2.9))^2$$

Psst! Remember:
these are the
Observed Heights...



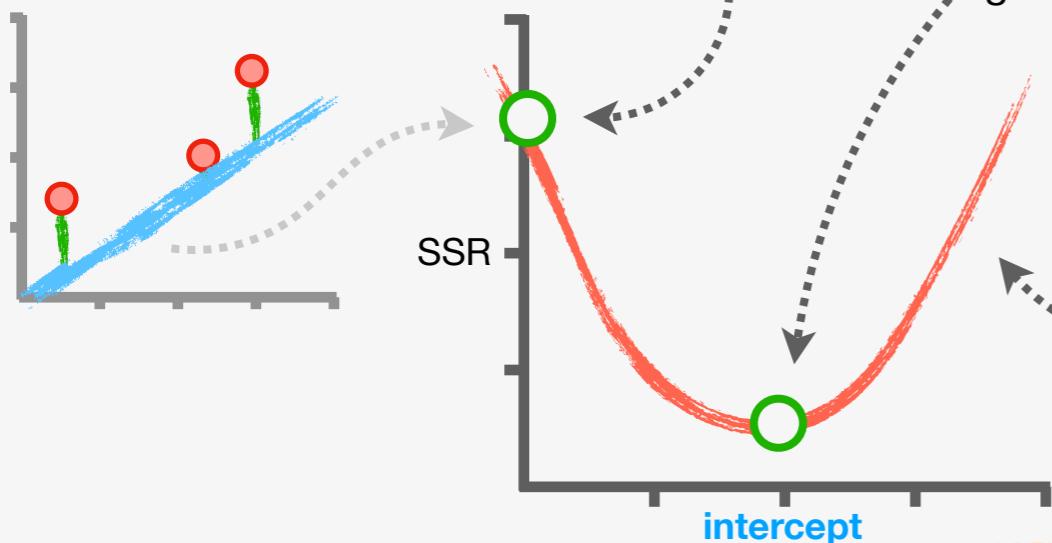
Gradient Descent: Details Part 5

12

Now, when we started with the y-axis **intercept = 0**, we got this **SSR**...

...so how do we take steps toward this y-axis intercept that gives us the lowest **SSR**...

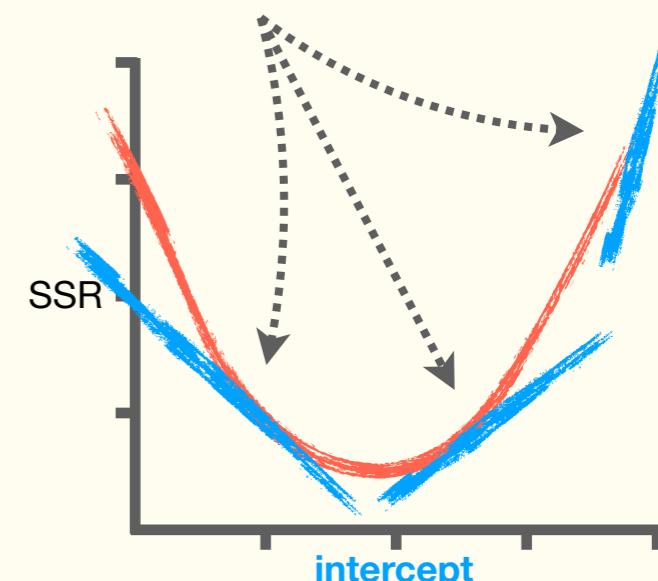
...and how do we know when to stop or if we've gone too far?



13

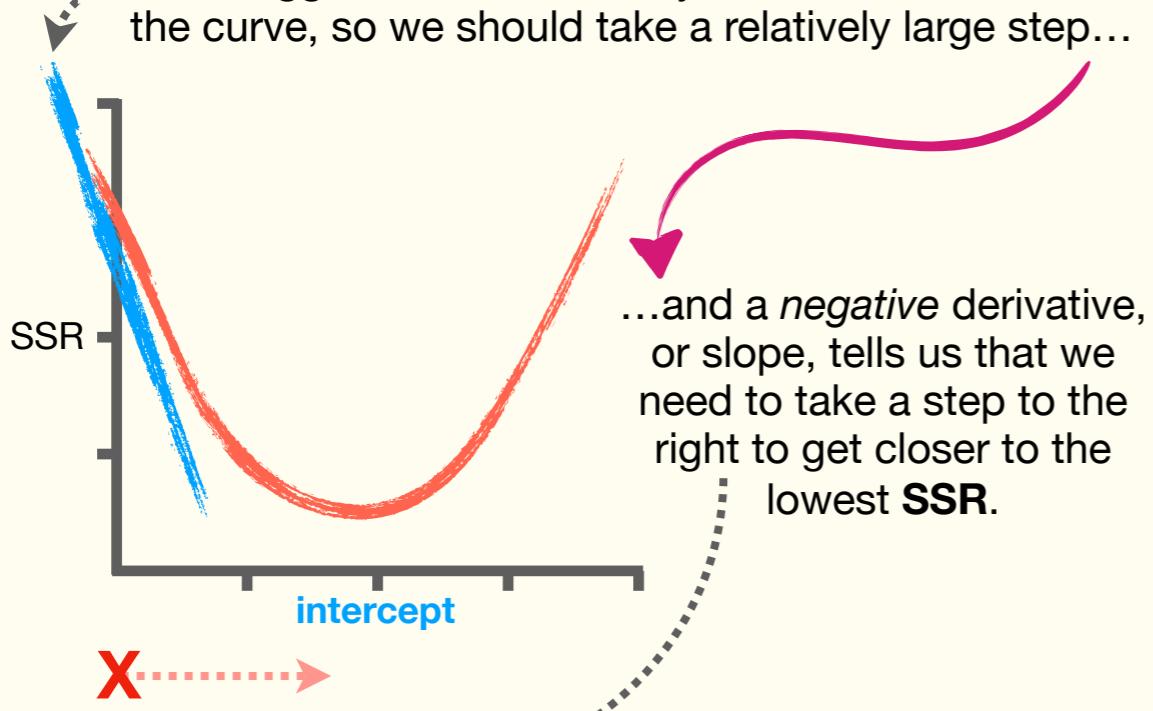
The answers to those questions come from the derivative of the curve, which tells us the slope of any **tangent line** that touches it.

NOTE: See **Appendix D** to learn more about derivatives.



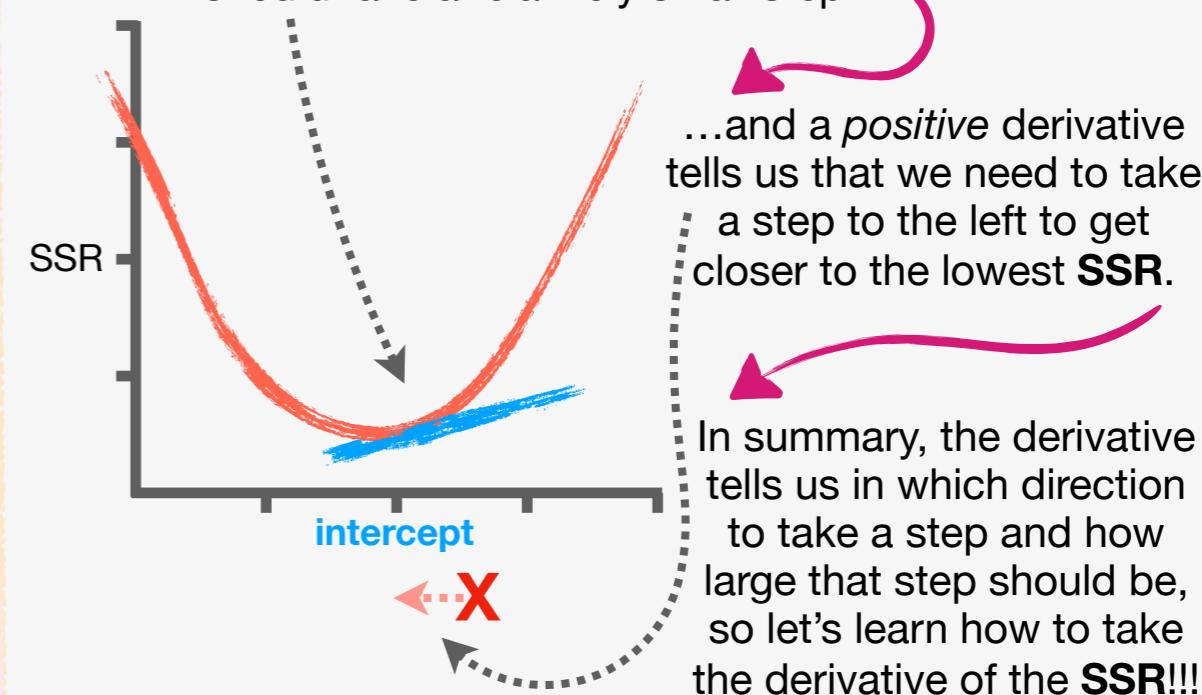
14

A relatively large value for the derivative, which corresponds to a relatively steep slope for the **tangent line**, suggests we're relatively far from the bottom of the curve, so we should take a relatively large step...



15

A relatively small value for the derivative suggests we're relatively close to the bottom of the curve, so we should take a relatively small step...



In summary, the derivative tells us in which direction to take a step and how large that step should be, so let's learn how to take the derivative of the **SSR**!!!

Gradient Descent: Details Part 6

16

Because a single term of the SSR consists of a **Residual**...

...wrapped in parentheses and squared...

...one way to take the derivative of the **SSR** is to use **The Chain Rule** (see **Appendix F** if you need to refresh your memory about how **The Chain Rule** works).

$$\text{SSR} = (\text{Height} - (\text{intercept} + 0.64 \times \text{Weight}))^2$$

Step 1: Create a *link* between the **intercept** and the **SSR** by rewriting the **SSR** as the function of the **Residual**.

$$\text{SSR} = (\text{Residual})^2$$

$$\text{Residual} = \text{Height} - (\text{intercept} + 0.64 \times \text{Weight})$$

Step 2: Because the **Residual** links the **intercept** to the **SSR**, **The Chain Rule** tells us that the derivative of the **SSR** with respect to the **intercept** is...

$$\frac{d \text{SSR}}{d \text{intercept}} = \frac{d \text{SSR}}{d \text{Residual}} \times \frac{d \text{Residual}}{d \text{intercept}}$$

Because of the subtraction, we can remove the parentheses by multiplying everything inside by **-1**.

Step 3: Use **The Power Rule** (**Appendix E**) to solve for the two derivatives.

$$\frac{d \text{SSR}}{d \text{Residual}} = \frac{d}{d \text{Residual}} (\text{Residual})^2 = 2 \times \text{Residual}$$

$$\begin{aligned} \frac{d \text{Residual}}{d \text{intercept}} &= \frac{d}{d \text{intercept}} \text{Height} - (\text{intercept} + 0.64 \times \text{Weight}) \\ &= \frac{d}{d \text{intercept}} \text{Height} - \text{intercept} - 0.64 \times \text{Weight} \\ &= 0 - 1 - 0 = -1 \end{aligned}$$

Step 4: Plug the derivatives into **The Chain Rule** to get the final derivative of the **SSR** with respect to the **intercept**.

$$\begin{aligned} \frac{d \text{SSR}}{d \text{intercept}} &= \frac{d \text{SSR}}{d \text{Residual}} \times \frac{d \text{Residual}}{d \text{intercept}} \\ &= 2 \times \text{Residual} \times -1 \\ &= 2 \times (\text{Height} - (\text{intercept} + 0.64 \times \text{Weight})) \times -1 \\ &= -2 \times (\text{Height} - (\text{intercept} + 0.64 \times \text{Weight})) \end{aligned}$$

Because the first and last terms do not include the **intercept**, their derivatives, with respect to the **intercept**, are both **0**. However, the second term is the negative **intercept**, so its derivative is **-1**.

Multiply this **-1** on the right by the **2** on the left to get **-2**.

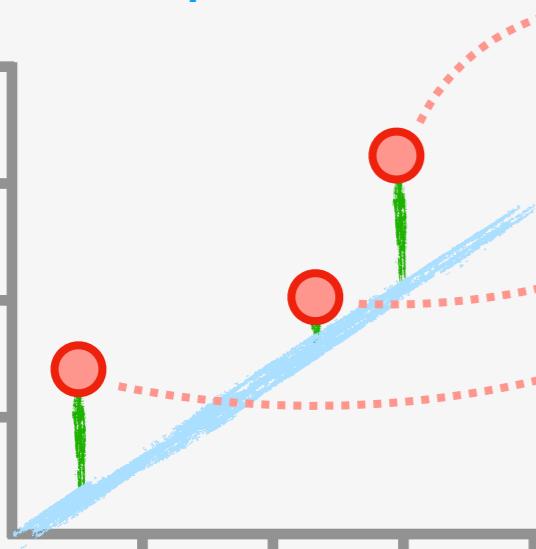
BAM!!!

Gradient Descent: Details Part 7

17

So far, we've calculated the derivative of the **SSR** for a single observation.

$$\begin{aligned} \text{SSR} &= (\text{Height} - (\text{intercept} + 0.64 \times \text{Weight}))^2 \\ \frac{d \text{SSR}}{d \text{intercept}} &= -2 \times (\text{Height} - (\text{intercept} + 0.64 \times \text{Weight})) \end{aligned}$$



However, we have three observations in the dataset, so the **SSR** and its derivative both have three terms.

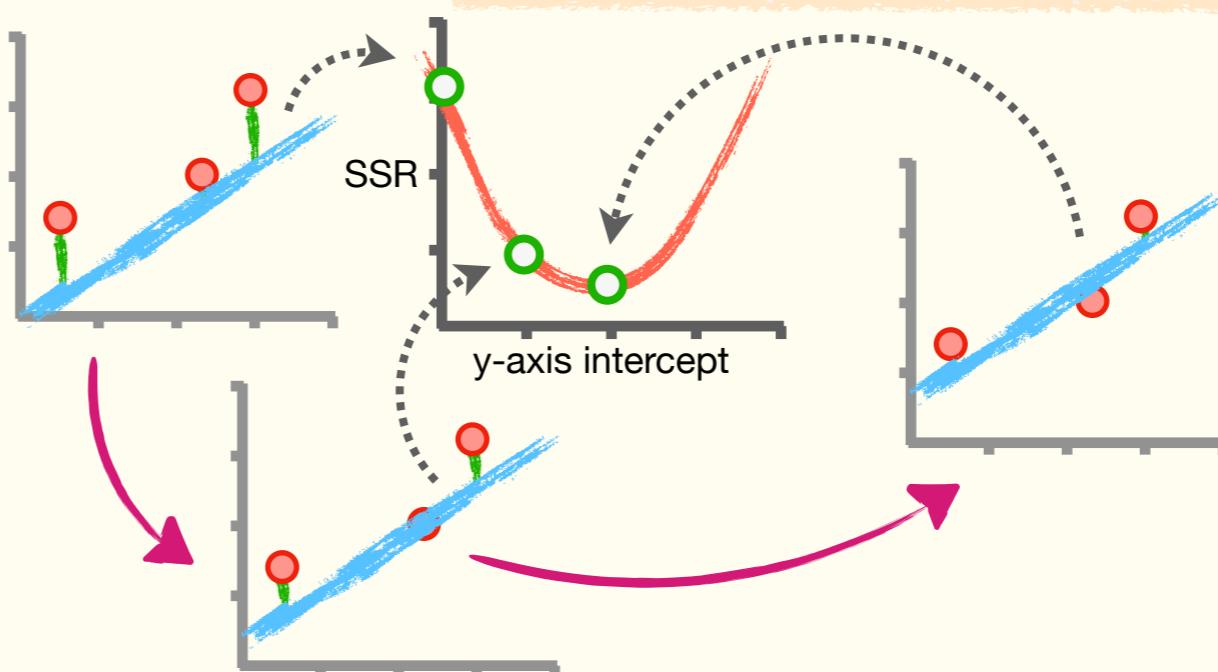
$$\begin{aligned} \text{SSR} &= (\text{Height} - (\text{intercept} + 0.64 \times \text{Weight}))^2 \\ &\quad + (\text{Height} - (\text{intercept} + 0.64 \times \text{Weight}))^2 \\ &\quad + (\text{Height} - (\text{intercept} + 0.64 \times \text{Weight}))^2 \end{aligned}$$

$$\begin{aligned} \frac{d \text{SSR}}{d \text{intercept}} &= -2 \times (\text{Height} - (\text{intercept} + 0.64 \times \text{Weight})) \\ &\quad + -2 \times (\text{Height} - (\text{intercept} + 0.64 \times \text{Weight})) \\ &\quad + -2 \times (\text{Height} - (\text{intercept} + 0.64 \times \text{Weight})) \end{aligned}$$

Gentle Reminder: Because we're using **Linear Regression** as our example, we don't actually **need** to use **Gradient Descent** to find the optimal value for the intercept. Instead, we could just set the derivative equal to **0** and solve for the **intercept**. This would be an analytical solution. However, by applying **Gradient Descent** to this problem, we can compare the optimal value that it gives us to the analytical solution and evaluate how well **Gradient Descent** performs. This will give us more confidence in **Gradient Descent** when we use it in situations without analytical solutions like **Logistic Regression** and **Neural Networks**.

18

Now that we have the derivative of the **SSR** for all 3 data points, we can go through, step-by-step, how **Gradient Descent** uses this derivative to find the **intercept** value that minimizes the **SSR**. However, before we get started, it's time for the dreaded **Terminology Alert!!!**



Terminology Alert!!! Parameters

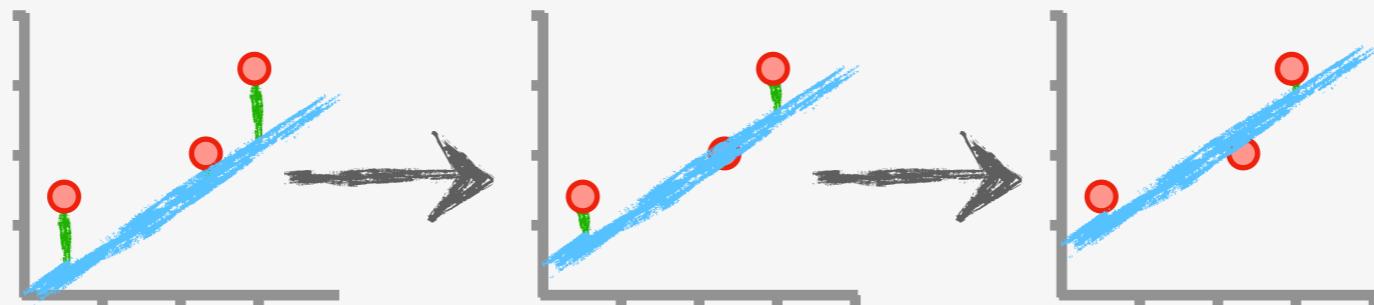
OH NO!!! MORE TERMINOLOGY!!!

1

In the current example, we're trying to optimize the y-axis **intercept**.

In machine learning lingo, we call the things we want to optimize **parameters**. So, in this case, we would call the y-axis **intercept** a **parameter**.

$$\text{Predicted Height} = \text{intercept} + 0.64 \times \text{Weight}$$



2

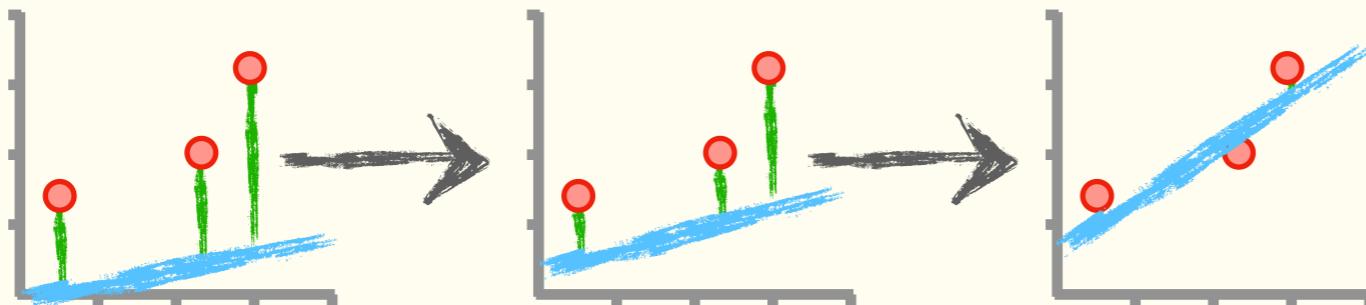
If we wanted to optimize both the y-axis **intercept** and the **slope**, then we would need to optimize two **parameters**.

tiny bam.

$$\text{Predicted Height} = \text{intercept} + \text{slope} \times \text{Weight}$$

3

Now that we know what we mean when we say **parameter**, let's see how **Gradient Descent** optimizes a single **parameter**, the **intercept**, one step at a time!!!



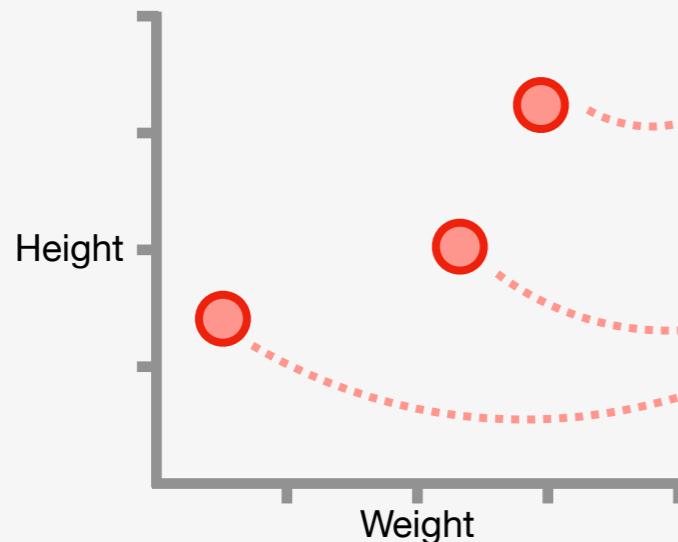
BAM!!!

Gradient Descent for One Parameter: Step-by-Step

1

First, plug the **Observed** values into the derivative of the **Loss** or **Cost Function**. In this example, the **SSR** is the **Loss** or **Cost Function**...

...so that means plugging the **Observed Weight** and **Height** measurements into the derivative of the **SSR**.



$$\frac{d \text{SSR}}{d \text{intercept}} = -2 \times (\text{Height} - (\text{intercept} + 0.64 \times \text{Weight}))$$

$$+ -2 \times (\text{Height} - (\text{intercept} + 0.64 \times \text{Weight}))$$

$$+ -2 \times (\text{Height} - (\text{intercept} + 0.64 \times \text{Weight}))$$

$$\frac{d \text{SSR}}{d \text{intercept}} = -2 \times (3.2 - (\text{intercept} + 0.64 \times 2.9))$$

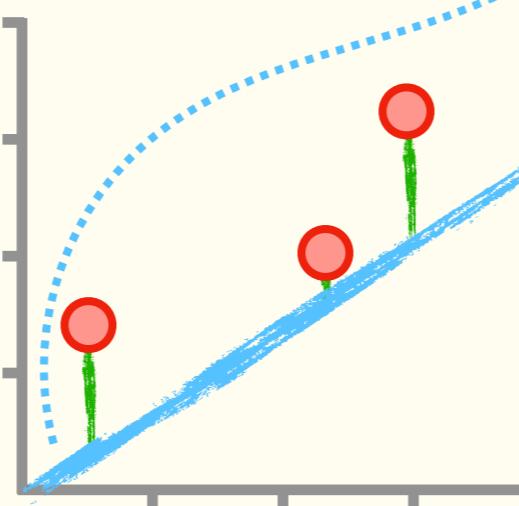
$$+ -2 \times (1.9 - (\text{intercept} + 0.64 \times 2.3))$$

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

2

Now we initialize the parameter we want to optimize with a random value. In this example, where we just want to optimize the y-axis **intercept**, we start by setting it to **0**.

$$\begin{aligned} \text{Height} &= \text{intercept} + 0.64 \times \text{Weight} \\ &= 0 + 0.64 \times \text{Weight} \end{aligned}$$

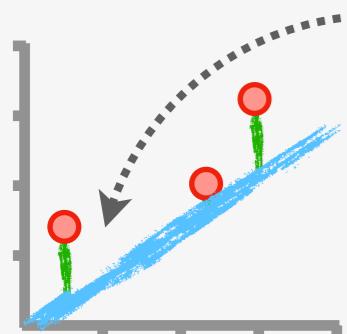


$$\begin{aligned} \frac{d \text{SSR}}{d \text{intercept}} &= -2 \times (3.2 - (0 + 0.64 \times 2.9)) \\ &+ -2 \times (1.9 - (0 + 0.64 \times 2.3)) \\ &+ -2 \times (1.4 - (0 + 0.64 \times 0.5)) \end{aligned}$$

Gradient Descent for One Parameter: Step-by-Step

3

Now evaluate the derivative at the current value for the **intercept**. In this case, the current value is 0.

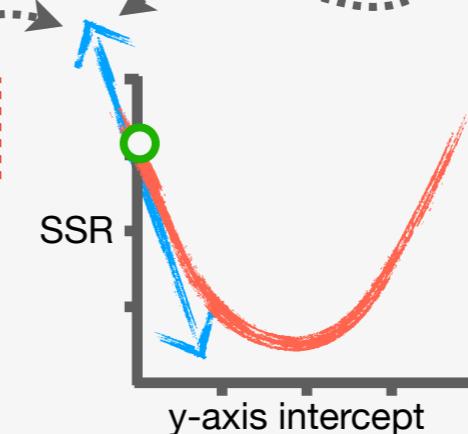


When we do the math, we get -5.7...

...thus, when the **intercept = 0**, the slope of this **tangent line** is -5.7.

$$\begin{aligned}\frac{d \text{SSR}}{d \text{intercept}} &= -2 \times (3.2 - (0 + 0.64 \times 2.9)) \\ &+ -2 \times (1.9 - (0 + 0.64 \times 2.3)) \\ &+ -2 \times (1.4 - (0 + 0.64 \times 0.5))\end{aligned}$$

$$= -5.7$$



4

Now calculate the **Step Size** with the following equation:

Gentle Reminder: The magnitude of the derivative is proportional to how big of a step we should take toward the minimum. The sign (+/-) tells us which direction.

$$\text{Step Size} = \text{Derivative} \times \text{Learning Rate}$$

$$= -5.7 \times 0.1$$

$$= -0.57$$

NOTE: The **Learning Rate** prevents us from taking steps that are too big and skipping past the lowest point in the curve. Typically, for **Gradient Descent**, the **Learning Rate** is determined automatically: it starts relatively large and gets smaller with every step taken. However, you can also use **Cross Validation** to determine a good value for the **Learning Rate**. In this case, we're setting the **Learning Rate to 0.1**.

5

Take a step from the **current intercept** to get closer to the optimal value with the following equation:

$$\text{New intercept} = \text{Current intercept} - \text{Step Size}$$

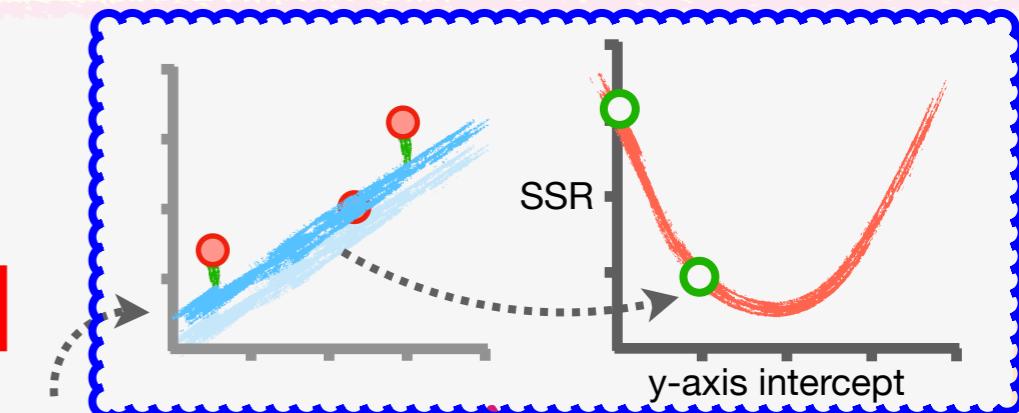
Remember, in this case, the **current intercept** is 0.

$$= 0 - (-0.57)$$

$$= 0.57$$

The **new intercept**, 0.57, moves the line up a little closer to the data...

...and it results in a lower **SSR**. Bam!



Gradient Descent for One Parameter: Step-by-Step

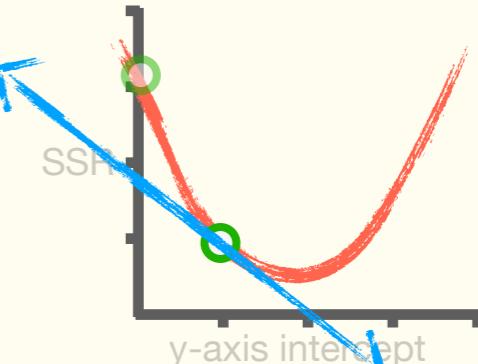
6

Now repeat the previous three steps, updating the **intercept** after each iteration until the **Step Size** is close to **0** or we take the maximum number of steps, which is often set to **1,000** iterations.

a Evaluate the derivative at the current value for the intercept...

$$\begin{aligned} \frac{d \text{SSR}}{d \text{intercept}} &= -2 \times (3.2 - (0.57 + 0.64 \times 2.9)) \\ &+ -2 \times (1.9 - (0.57 + 0.64 \times 2.3)) \\ &+ -2 \times (1.4 - (0.57 + 0.64 \times 0.5)) \end{aligned}$$

$$= -2.3$$



b Calculate the **Step Size**...

$$\text{Step Size} = \text{Derivative} \times \text{Learning Rate}$$

$$= -2.3 \times 0.1$$

$$= -0.23$$

NOTE: The **Step Size** is smaller than before because the slope of the **tangent line** is not as steep as before. The smaller slope means we're getting closer to the optimal value.

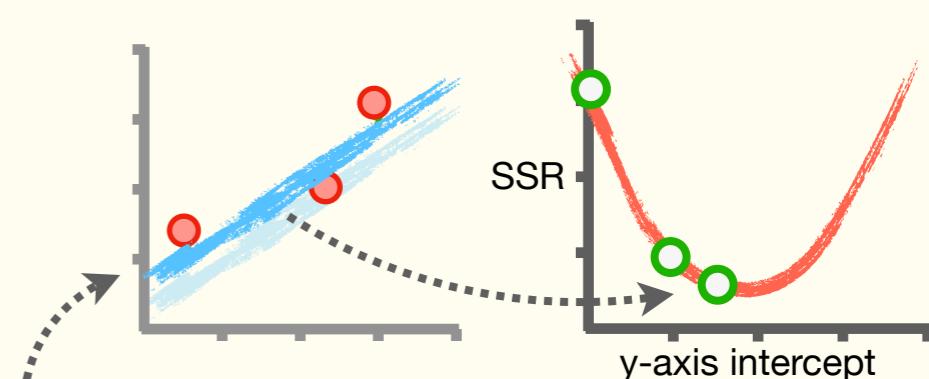
c Calculate the **new intercept** value...

$$\text{New intercept} = \text{Current intercept} - \text{Step Size}$$

$$= 0.57 - (-0.23)$$

$$= 0.8$$

The **new intercept**, 0.8, moves the line up a little closer to the data...



...and it results in a lower **SSR**. Double Bam!

Gradient Descent for One Parameter: Step-by-Step

7

After 7 iterations...

a

Evaluate the derivative at
the current value...

b

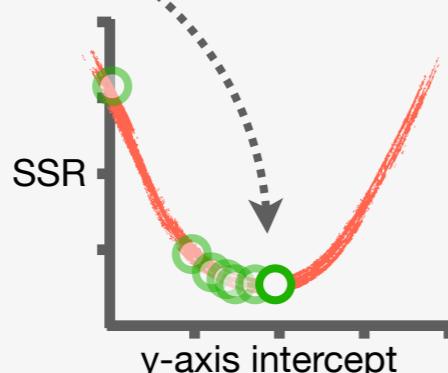
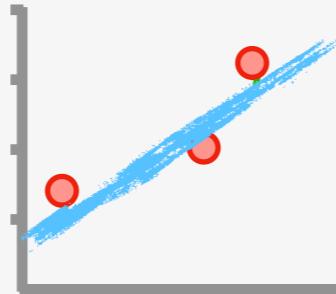
Calculate the
Step Size...

c

Calculate the
new value...

...the **Step Size** was very close
to 0, so we stopped with the
current intercept = 0.95...

...and we made it to
the lowest **SSR**.



8

If, earlier on, instead of using **Gradient Descent**, we simply set the derivative to 0 and solved for the **intercept**, we would have gotten 0.95, which is the same value that **Gradient Descent** gave us. Thus, **Gradient Descent** did a decent job.

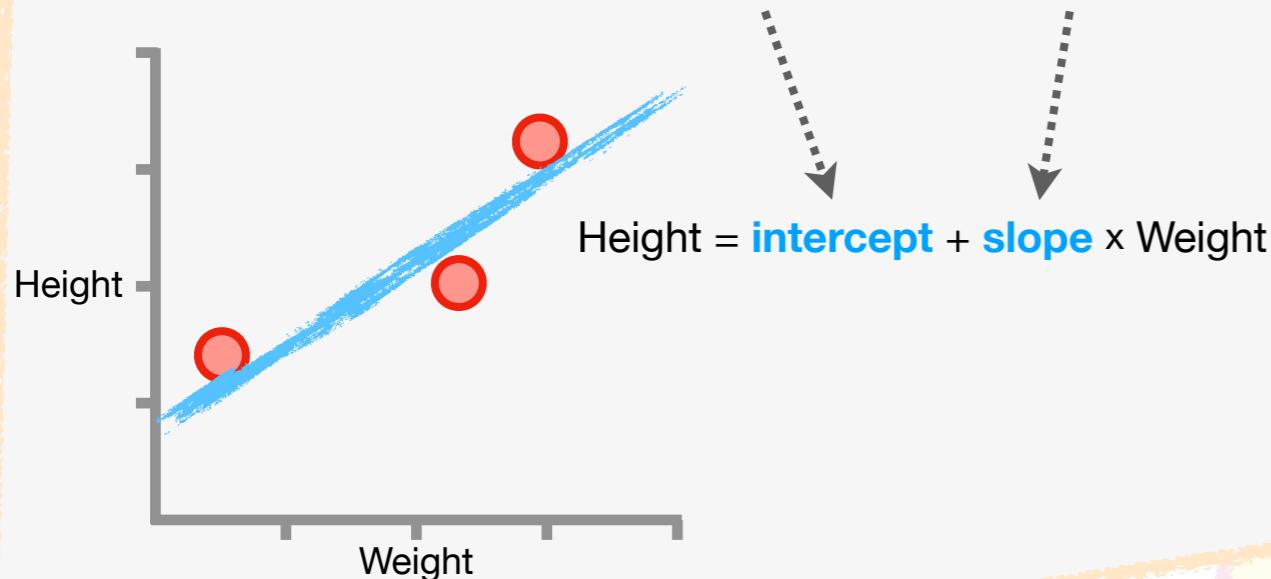
BAM???

Not yet! Now let's see how well
Gradient Descent optimizes the
intercept and the **slope**!

Optimizing Two or More Parameters: Details

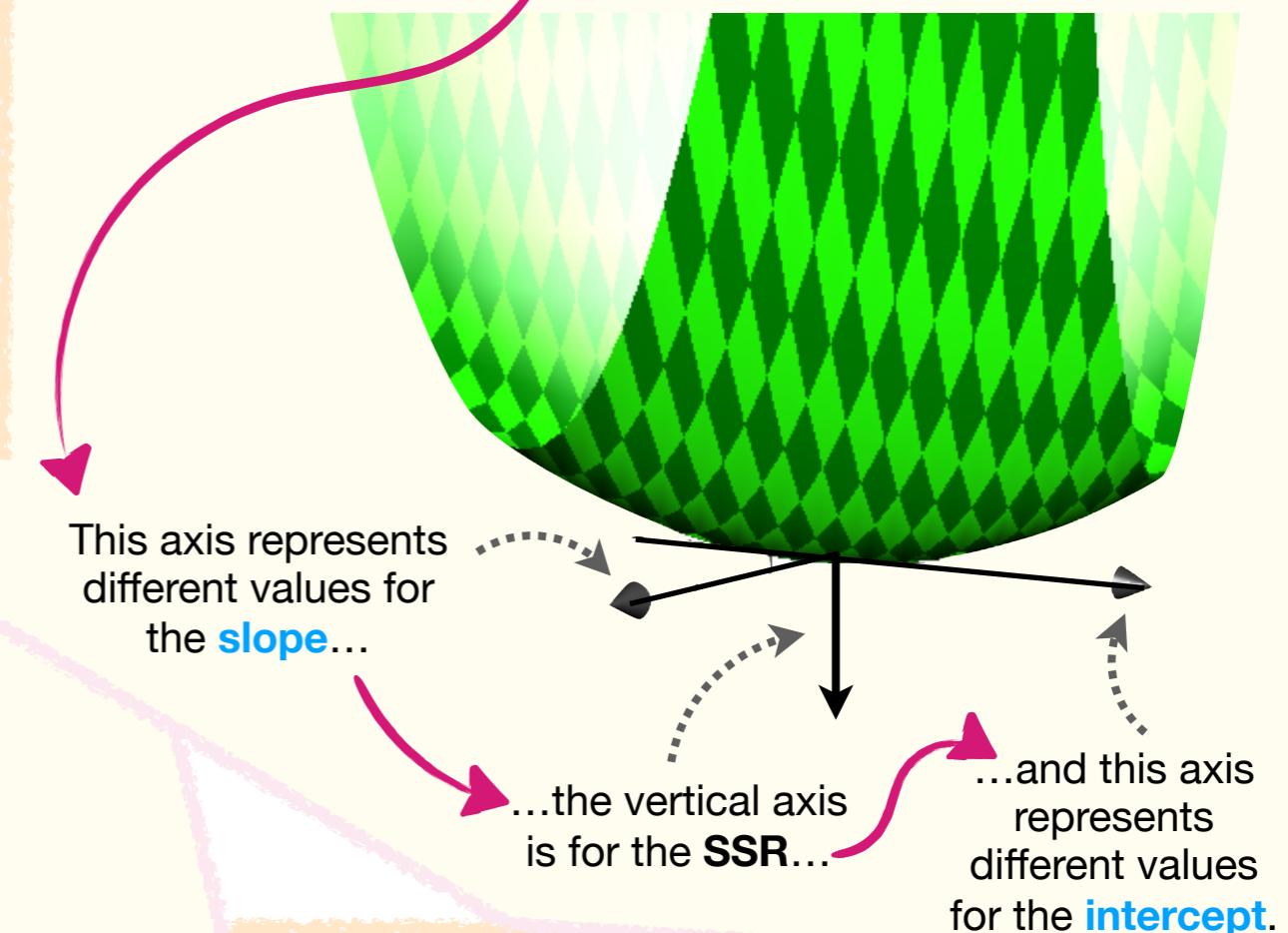
1

Now that we know how to optimize the **intercept** of the line that minimizes the **SSR**, let's optimize both the **intercept** and the **slope**.



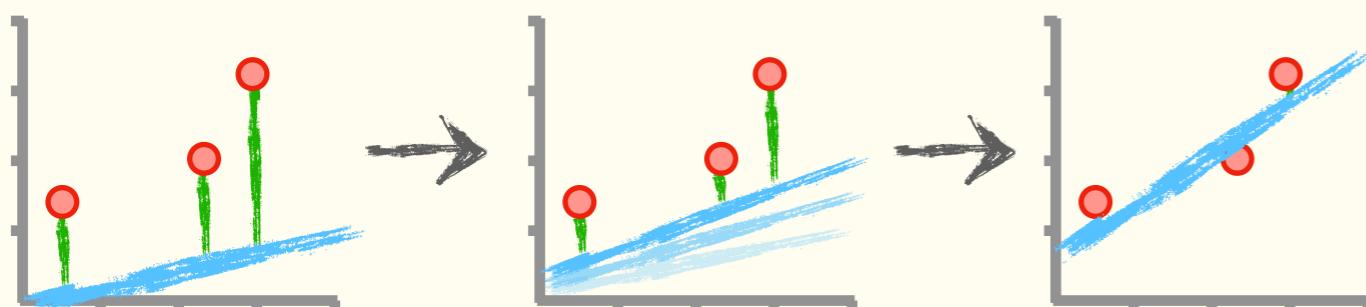
2

When we optimize two parameters, we get a 3-dimensional graph of the **SSR**.



3

Just like before, the goal is to find the parameter values that give us the lowest **SSR**. And just like before, **Gradient Descent** initializes the parameters with random values and then uses derivatives to update those parameters, one step at a time, until they're optimal.



4

So, now let's learn how to take derivatives of the **SSR** with respect to both the **intercept** and the **slope**.

$$\text{SSR} = (\text{Height} - (\text{intercept} + \text{slope} \times \text{Weight}))^2$$

Taking Multiple (Partial) Derivatives of the SSR: Part 1

1

The good news is that taking the derivative of the **SSR** with respect to the **intercept** is exactly the same as before.

$$\text{SSR} = (\text{Height} - (\text{intercept} + \text{slope} \times \text{Weight}))^2$$

We can use **The Chain Rule** to tell us how the **SSR** changes with respect to the **intercept**.

Step 1: Create a *link* between the **intercept** and the **SSR** by rewriting the **SSR** as the function of the **Residual**.

Step 2: Because the **Residual** links the **intercept** to the **SSR**, **The Chain Rule** tells us that the derivative of the **SSR** with respect to the **intercept** is...

Step 3: Use **The Power Rule** to solve for the two derivatives.

$$\frac{d \text{SSR}}{d \text{Residual}} = \frac{d}{d \text{Residual}} (\text{Residual})^2 = 2 \times \text{Residual}$$

$$\frac{d \text{SSR}}{d \text{intercept}} = \frac{d \text{SSR}}{d \text{Residual}} \times \frac{d \text{Residual}}{d \text{intercept}}$$

Because of the subtraction, we can remove the parentheses by multiplying everything inside by **-1**.

$$\begin{aligned}\frac{d \text{Residual}}{d \text{intercept}} &= \frac{d}{d \text{intercept}} \text{Height} - (\text{intercept} + \text{slope} \times \text{Weight}) \\ &= \frac{d}{d \text{intercept}} \text{Height} - \text{intercept} - \text{slope} \times \text{Weight} \\ &= 0 - 1 - 0 = -1\end{aligned}$$

Step 4: Plug the derivatives into **The Chain Rule** to get the final derivative of the **SSR** with respect to the **intercept**.

BAM!!!

$$\begin{aligned}\frac{d \text{SSR}}{d \text{intercept}} &= \frac{d \text{SSR}}{d \text{Residual}} \times \frac{d \text{Residual}}{d \text{intercept}} = 2 \times \text{Residual} \times -1 \\ &= 2 \times (\text{Height} - (\text{intercept} + \text{slope} \times \text{Weight})) \times -1 \\ &= -2 \times (\text{Height} - (\text{intercept} + \text{slope} \times \text{Weight}))\end{aligned}$$

Multiply this **-1** on the right by the **2** on the left to get **-2**.

Because the first and last terms do not include the **intercept**, their derivatives, with respect to the **intercept**, are both **0**. However, the second term is the negative **intercept**, so its derivative is **-1**.

Taking Multiple (Partial) Derivatives of the SSR: Part 2

2

The other good news is that taking the derivative of the **SSR** with respect to the **slope** is very similar to what we just did for the **intercept**.

$$\text{SSR} = (\text{Height} - (\text{intercept} + \text{slope} \times \text{Weight}))^2$$

Step 1: Create a *link* between the **slope** and the **SSR** by rewriting the **SSR** as the function of the **Residual**.

$$\text{SSR} = (\text{Residual})^2$$

We can use **The Chain Rule** to tell us how the **SSR** changes with respect to the **slope**.

NOTE: A collection of derivatives of the same function but with respect to different parameters is called a **Gradient**, so this is where **Gradient Descent** gets its name from. We'll use the *gradient* to descend to the lowest **SSR**.

Step 2: Because the **Residual** links the **slope** to the **SSR**, **The Chain Rule** tells us that the derivative of the **SSR** with respect to the **slope** is...

$$\frac{d \text{SSR}}{d \text{slope}} = \frac{d \text{SSR}}{d \text{Residual}} \times \frac{d \text{Residual}}{d \text{slope}}$$

Step 3: Use **The Power Rule** to solve for the two derivatives.

$$f'(x^n) = nx^{n-1}$$

$$\frac{d}{dx} x^n = nx^{n-1}$$

$$\frac{d \text{SSR}}{d \text{Residual}} = \frac{d}{d \text{Residual}} (\text{Residual})^2 = 2 \times \text{Residual}$$

Because of the subtraction, we can remove the parentheses by multiplying everything inside by **-1**.

$$\begin{aligned} \frac{d \text{Residual}}{d \text{slope}} &= \frac{d}{d \text{slope}} \text{Height} - (\text{intercept} + \text{slope} \times \text{Weight}) \\ &= \frac{d}{d \text{slope}} \text{Height} - \text{intercept} - \text{slope} \times \text{Weight} \\ &= 0 - 0 - \text{Weight} = -\text{Weight} \end{aligned}$$

Because the first and second terms do not include the **slope**, their derivatives, with respect to the **slope**, are both **0**. However, the last term is the negative **slope** times **Weight**, so its derivative is **-Weight**.

Step 4: Plug the derivatives into **The Chain Rule** to get the final derivative of the **SSR** with respect to the **slope**.

$$\frac{d \text{SSR}}{d \text{slope}} = \frac{d \text{SSR}}{d \text{Residual}} \times \frac{d \text{Residual}}{d \text{slope}} = 2 \times \text{Residual} \times -\text{Weight}$$

$$\begin{aligned} &= 2 \times (\text{Height} - (\text{intercept} + \text{slope} \times \text{Weight})) \times -\text{Weight} \\ &= -2 \times \text{Weight} \times (\text{Height} - (\text{intercept} + \text{slope} \times \text{Weight})) \end{aligned}$$

Multiply this **-Weight** on the right by the **2** on the left to get **-2 × Weight**.

DOUBLE BAM!!!

Gradient Descent for Two Parameters: Step-by-Step

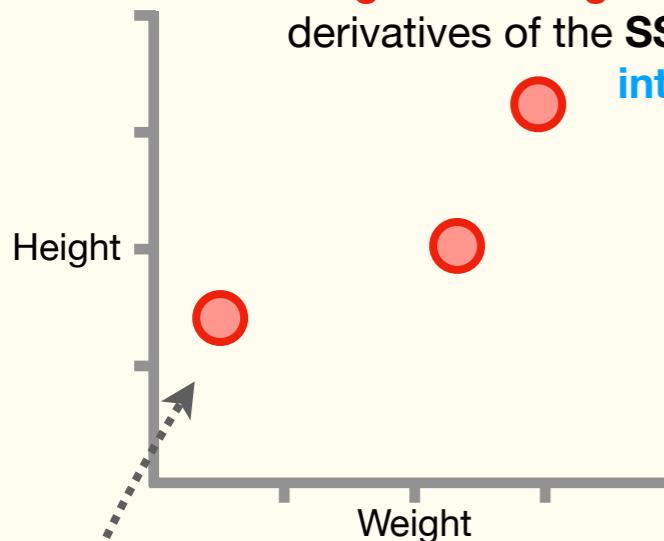
1

Plug the **Observed** values into the derivatives of the **Loss or Cost Function**. In this example, the **SSR** is the **Loss or Cost Function**, so we'll plug the **Observed Weight** and **Height** measurements into the two derivatives of the **SSR**, one with respect to the



intercept...

....and one with
respect to the slope.



$$\frac{d \text{SSR}}{d \text{intercept}}$$

$$= -2 \times (\text{Height}_1 - (\text{intercept} + \text{slope} \times \text{Weight}_1)) \\ + -2 \times (\text{Height}_2 - (\text{intercept} + \text{slope} \times \text{Weight}_2)) \\ + -2 \times (\text{Height}_3 - (\text{intercept} + \text{slope} \times \text{Weight}_3))$$

$$\frac{d \text{SSR}}{d \text{slope}}$$

$$= -2 \times \text{Weight}_1 \times (\text{Height}_1 - (\text{intercept} + \text{slope} \times \text{Weight}_1)) \\ + -2 \times \text{Weight}_2 \times (\text{Height}_2 - (\text{intercept} + \text{slope} \times \text{Weight}_2)) \\ + -2 \times \text{Weight}_3 \times (\text{Height}_3 - (\text{intercept} + \text{slope} \times \text{Weight}_3))$$

Gentle Reminder: The Weight and Height values that we're plugging into the derivatives come from the raw data in the graph.

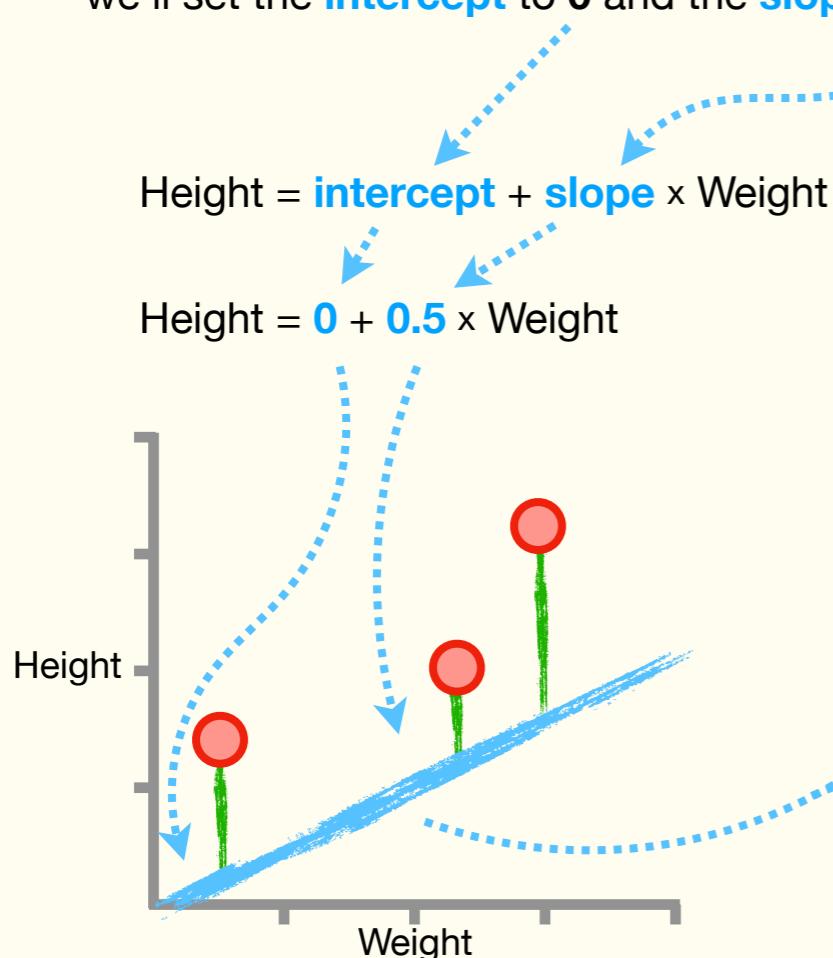
$$\frac{d \text{SSR}}{d \text{slope}}$$

$$= -2 \times 2.9 (3.2 - (\text{intercept} + \text{slope} \times 2.9)) \\ + -2 \times 2.3 (1.9 - (\text{intercept} + \text{slope} \times 2.3)) \\ + -2 \times 0.5 (1.4 - (\text{intercept} + \text{slope} \times 0.5))$$

Gradient Descent for Two Parameters: Step-by-Step

2

Now initialize the parameter, or parameters, that we want to optimize with random values. In this example, we'll set the **intercept** to **0** and the **slope** to **0.5**.



$$\frac{d \text{SSR}}{d \text{intercept}} = -2 \times (3.2 - (\text{intercept} + \text{slope} \times 2.9)) \\ + -2 \times (1.9 - (\text{intercept} + \text{slope} \times 2.3)) \\ + -2 \times (1.4 - (\text{intercept} + \text{slope} \times 0.5))$$

$$\frac{d \text{SSR}}{d \text{intercept}} = -2 \times (3.2 - (0 + 0.5 \times 2.9)) \\ + -2 \times (1.9 - (0 + 0.5 \times 2.3)) \\ + -2 \times (1.4 - (0 + 0.5 \times 0.5))$$

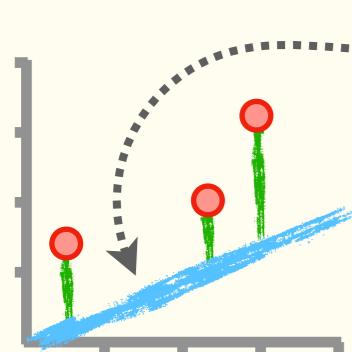
$$\frac{d \text{SSR}}{d \text{slope}} = -2 \times 2.9 \times (3.2 - (\text{intercept} + \text{slope} \times 2.9)) \\ + -2 \times 2.3 \times (1.9 - (\text{intercept} + \text{slope} \times 2.3)) \\ + -2 \times 0.5 \times (1.4 - (\text{intercept} + \text{slope} \times 0.5))$$

$$\frac{d \text{SSR}}{d \text{slope}} = -2 \times 2.9 \times (3.2 - (0 + 0.5 \times 2.9)) \\ + -2 \times 2.3 \times (1.9 - (0 + 0.5 \times 2.3)) \\ + -2 \times 0.5 \times (1.4 - (0 + 0.5 \times 0.5))$$

Gradient Descent for Two Parameters: Step-by-Step

3

Evaluate the derivatives at the current values for the **intercept**, 0, and **slope**, 0.5.



$$\begin{aligned} \frac{d \text{SSR}}{d \text{intercept}} &= -2 \times (3.2 - (0 + 0.5 \times 2.9)) \\ &\quad + -2 \times (1.9 - (0 + 0.5 \times 2.3)) \\ &\quad + -2 \times (1.4 - (0 + 0.5 \times 0.5)) \end{aligned}$$

= -7.3

$$\begin{aligned} \frac{d \text{SSR}}{d \text{slope}} &= -2 \times 2.9 \times (3.2 - (0 + 0.5 \times 2.9)) \\ &\quad + -2 \times 1.9 \times (2.3 - (0 + 0.5 \times 1.9)) \\ &\quad + -2 \times 0.5 \times (1.4 - (0 + 0.5 \times 0.5)) \end{aligned}$$

= -14.8

4

Calculate the **Step Sizes**: one for the **intercept**...

$$\begin{aligned} \text{Step Size}_{\text{Intercept}} &= \text{Derivative} \times \text{Learning Rate} \\ &= -7.3 \times 0.01 \\ &= -0.073 \end{aligned}$$

...and one for the **slope**.

$$\begin{aligned} \text{Step Size}_{\text{Slope}} &= \text{Derivative} \times \text{Learning Rate} \\ &= -14.8 \times 0.01 \\ &= -0.148 \end{aligned}$$

NOTE: We're using a smaller **Learning Rate** now (0.01) than before (0.1) because **Gradient Descent** can be very sensitive to it. However, as we said earlier, usually the **Learning Rate** is determined automatically.

5

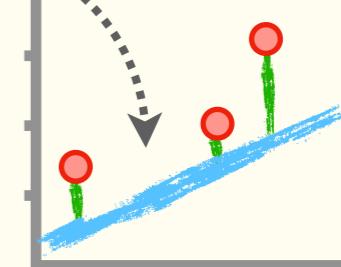
Take a step from the **current intercept**, 0, and **slope**, 0.5, to get closer to the optimal values...

$$\begin{aligned} \text{New intercept} &= \text{Current intercept} - \text{Step Size}_{\text{Intercept}} \\ &= 0 - (-0.073) \\ &= 0.073 \end{aligned}$$

...and the **intercept** increases from 0 to 0.073, the **slope** increases from 0.5 to 0.648, and the **SSR** decreases. **BAM!**

$$\begin{aligned} \text{New slope} &= \text{Current slope} - \text{Step Size}_{\text{Slope}} \\ &= 0.5 - (-0.148) \end{aligned}$$

= 0.648



Gradient Descent for Two Parameters: Step-by-Step

6

And after 475 iterations...

a

Evaluate the derivatives at their current values...

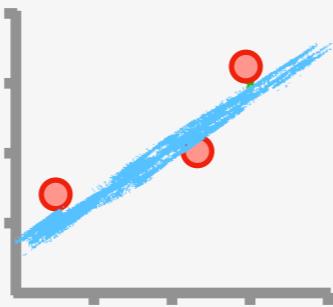
b

Calculate the Step Sizes...

c

Calculate the new values...

...the **Step Size** was very close to **0**, so we stopped with the **current intercept** = **0.95** and the **current slope** = **0.64**...



...and we made it to the lowest **SSR**.

This axis represents different values for the **slope**...

...this axis is for the **SSR**...

...and this axis represents different values for the **intercept**.

7

If, earlier on, instead of using **Gradient Descent**, we simply set the derivatives to **0** and solved for the **intercept** and **slope**, we would have gotten **0.95** and **0.64**, which are the same values **Gradient Descent** gave us. Thus, **Gradient Descent** did a great job, and we can confidently use it in situations where there are no analytical solutions, like **Logistic Regression** and **Neural Networks**.

TRIPLE BAM!!!

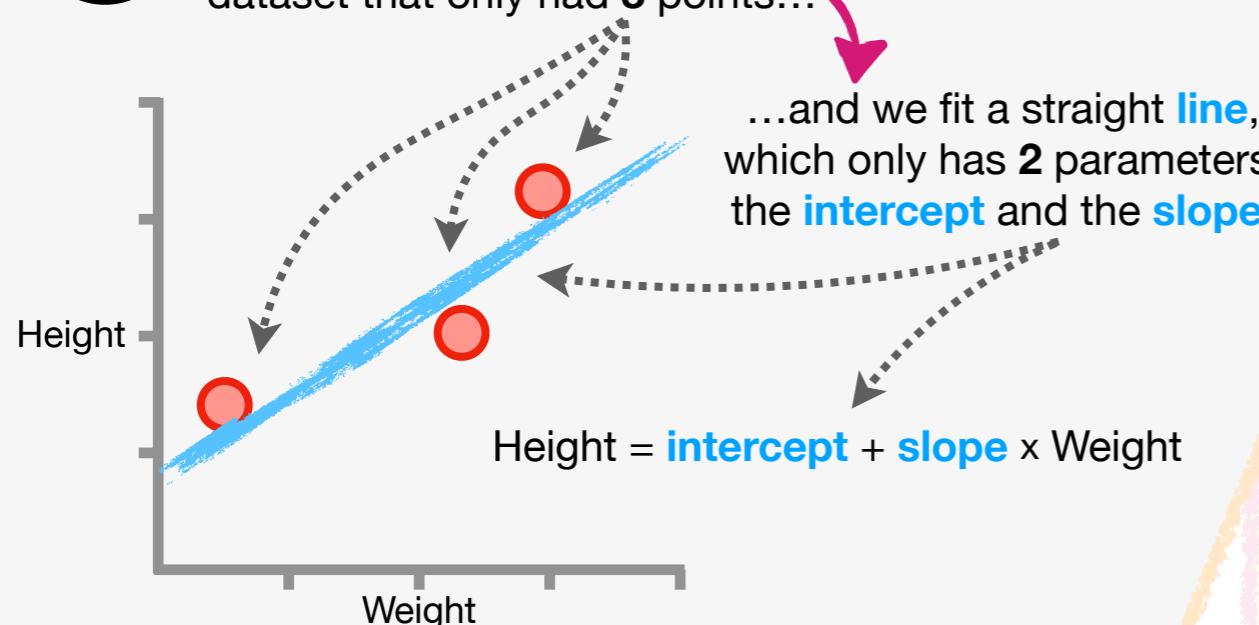
Gradient Descent is awesome, but when we have a lot of data or a lot of parameters, it can be slow. Is there any way to make it faster?

YES! Read on to learn about **Stochastic Gradient Descent**.

Stochastic Gradient Descent: Main Ideas

1

So far, things have been pretty simple. We started out with a tiny dataset that only had **3** points...



2

Because there were only **2** parameters, we only had **2** derivatives that we had to calculate in each step...

$$\frac{d \text{SSR}}{d \text{intercept}}$$

$$\frac{d \text{SSR}}{d \text{slope}}$$

...and because we only had **3** data points, each derivative only needed to compute **3** terms per derivative.

$$\begin{aligned} \frac{d \text{SSR}}{d \text{intercept}} &= -2 \times (3.2 - (\text{intercept} + \text{slope} \times 2.9)) \\ &\quad + -2 \times (1.9 - (\text{intercept} + \text{slope} \times 2.3)) \\ &\quad + -2 \times (1.4 - (\text{intercept} + \text{slope} \times 0.5)) \end{aligned}$$

$$\begin{aligned} \frac{d \text{SSR}}{d \text{slope}} &= -2 \times 2.9 \times (3.2 - (\text{intercept} + \text{slope} \times 2.9)) \\ &\quad + -2 \times 2.3 \times (1.9 - (\text{intercept} + \text{slope} \times 2.3)) \\ &\quad + -2 \times 0.5 \times (1.4 - (\text{intercept} + \text{slope} \times 0.5)) \end{aligned}$$

3

However, what if we had **1,000,000** data points? Then we would have to compute **1,000,000** terms per derivative.

Ugh!

And what if we had a more complicated model with **10,000** parameters? Then we would have **10,000** derivatives to compute.

Double Ugh!

Taking **10,000** derivatives, each with **1,000,000** terms to compute, is a lot of work, and all of that work only gets us one step into the process that can take **1,000s** of steps!!!

TRIPLE UGH!

Thus, for **BIG DATA**, **Gradient Descent** requires a lot of computation and can be slow.

4

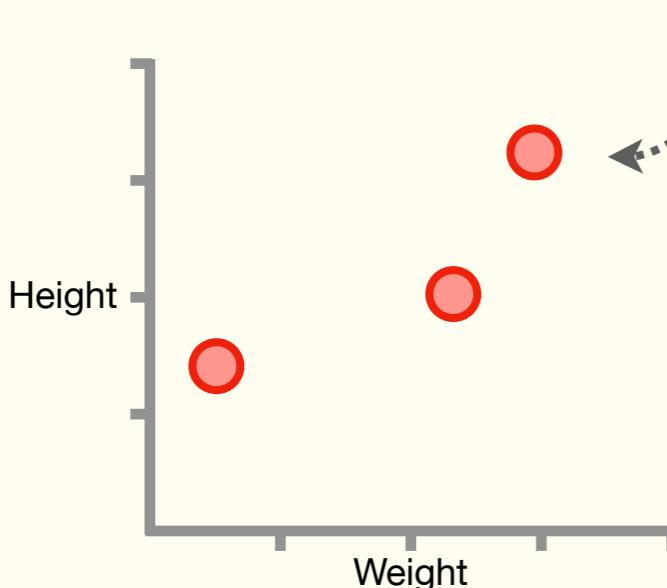
The good news is that **Stochastic Gradient Descent** can drastically reduce the amount of computation required to optimize parameters. Although it sounds fancy, the word **Stochastic** just means **Randomly Determined** and all **Stochastic Gradient Descent** does is randomly select one data point per step. So, regardless of how large your dataset is, only one term is computed per derivative for each iteration.

BAM!

Stochastic Gradient Descent: Details Part 1

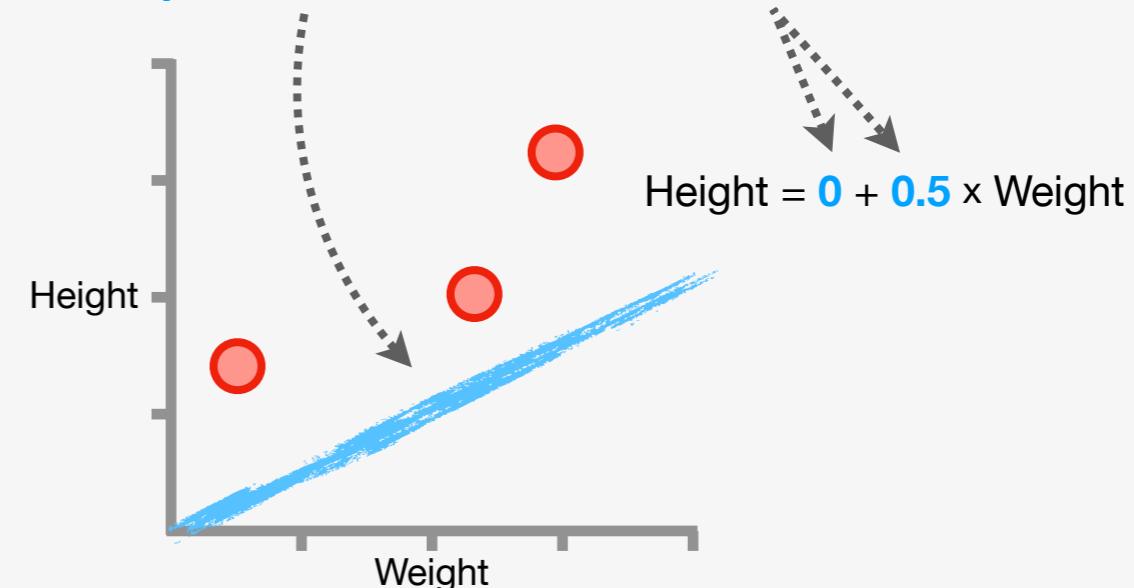
1

To see how **Stochastic Gradient Descent** works, let's go back to our simple example, where we want to fit a line to **3** data points.



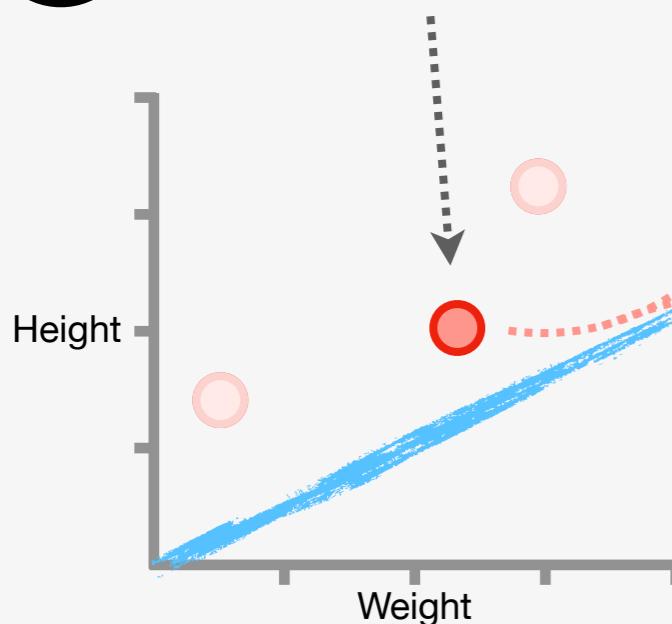
2

And just like with normal **Gradient Descent**, we start by initializing the **intercept** and **slope** of the line with random values.



3

Now we randomly pick one point. In this case, we'll pick this one in the middle.



4

Then, we evaluate the derivatives using just that single point...

$$\frac{d \text{SSR}}{d \text{intercept}} = -2 \times (\text{Height} - (\text{intercept} + \text{slope} \times \text{Weight}))$$

$$\frac{d \text{SSR}}{d \text{slope}} = -2 \times \text{Weight} \times (\text{Height} - (\text{intercept} + \text{slope} \times \text{Weight}))$$

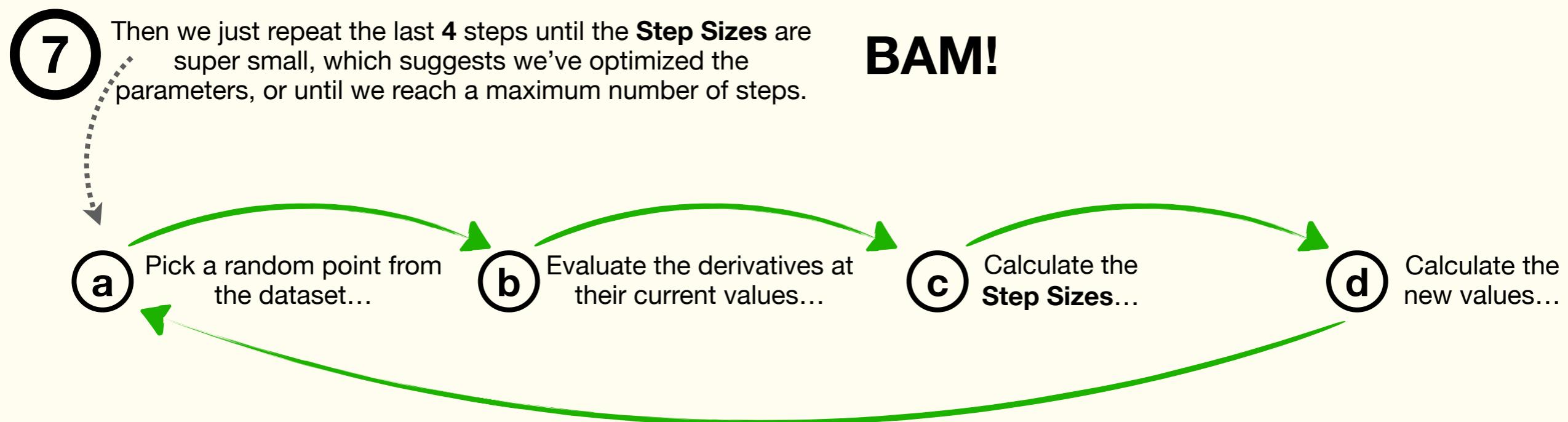
5

...and then we calculate the **Step Sizes**...

6

...and then we calculate the new values.

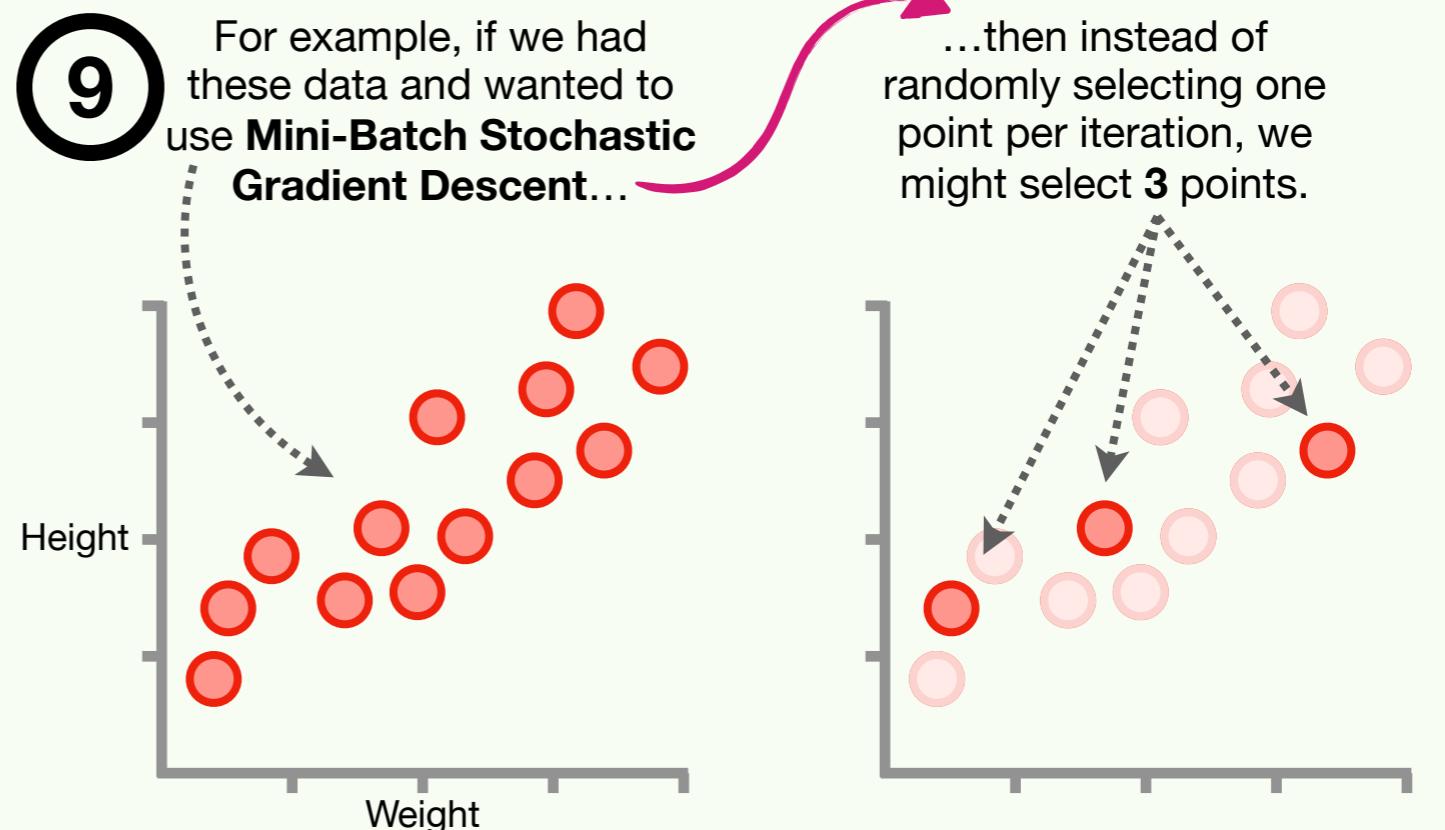
Stochastic Gradient Descent: Details Part 2



8 **TERMINOLOGY ALERT!!!**

Although a strict definition of **Stochastic Gradient Descent** says that we only select a single point per iteration, it's much more common to randomly select a small subset of the observations. This is called **Mini-Batch Stochastic Gradient Descent**.

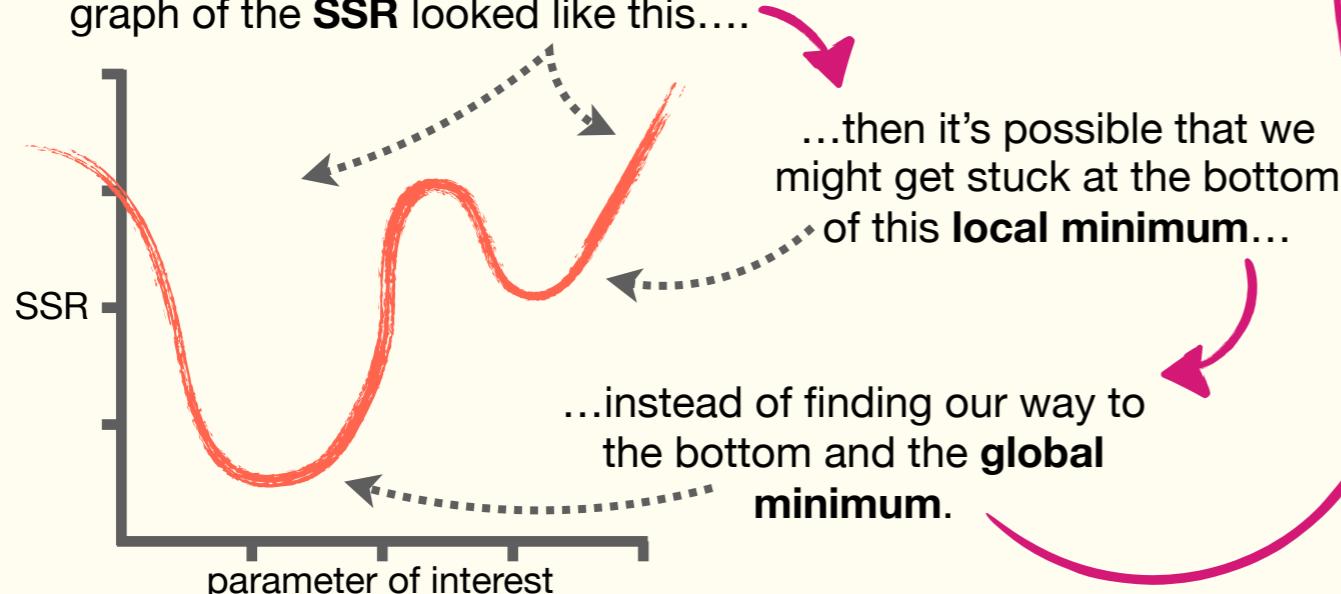
Descent. Using a small subset, rather than a single point, usually converges on the optimal values in fewer steps and takes much less time than using all of the data.



Gradient Descent: FAQ

Will Gradient Descent always find the best parameter values?

Unfortunately, **Gradient Descent** does not always find the best parameter values. For example, if the graph of the **SSR** looked like this....



When this happens (when we get stuck in a local minimum instead of finding the global minimum), it's a bummer. Even worse, usually it's not possible to graph the **SSR**, so we might not even know we're in one, of potentially many, local minimums. However, there are a few things we can do about it. We can:

- 1) Try again using different random numbers to initialize the parameters that we want to optimize. Starting with different values may avoid a local minimum.
- 2) Fiddle around with the **Step Size**. Making it a little larger may help avoid getting stuck in a local minimum.
- 3) Use **Stochastic Gradient Descent**, because the extra randomness helps avoid getting trapped in a local minimum.

How do you choose the size of a Mini-Batch for Stochastic Gradient Descent?

The answer to this question really depends on the computer hardware you're using to train (optimize) your model. For example, because one of the main reasons we use **Mini-Batch Stochastic Gradient Descent** is to train our model as quickly as possible, one major consideration is how much high-speed memory we have access to. The more high-speed memory we have, the larger the **Mini-Batch** can be.

Now let's talk about how to make classifications using **Logistic Regression**, which has no analytical solution and is often optimized with **Gradient Descent**.

