Chapter 3: Big Data Analytics and Machine Learning

Introduction

- Machine learning is about extracting knowledge from data.
- It is a research field at the intersection of statistics, artificial intelligence, and computer science
- also known as predictive analytics or statistical learning
- The application of machine learning methods has in recent years become ubiquitous in everyday life

Introduction

- Examples:
  - automatic recommendations of which movies to watch,
  - automatic recommendations what food to order
  - Recomending which products to buy,
  - personalized online radio and
  - recognizing your friends in your photos,
- many modern websites and devices have machine learning algorithms at their core.
- When you look at a complex website like Facebook, Amazon, or Netflix, it is very likely that every part of the site contains multiple machine learning models.

Why Machine Learning?

- In the early days of "intelligent" applications, many systems used handcoded rules of "if" and "else" decisions to process data or adjust to user input.
  - Eg spam filter: You could make up a blacklist of words that would result in an email being marked as spam
- Manually crafting decision rules is feasible for some applications, particularly those in which humans have a good understanding of the process to model.
- One example of where this hand coded approach will fail is in detecting faces in images
- Today, every smartphone can detect a face in an image.
- However, face detection was an unsolved problem until as recently as 2001.

Why Machine Learning?

- The main problem is that the way in which pixels (which make up an image in a computer) are "perceived" by the computer is very different from how humans perceive a face.
- This difference in representation makes it basically impossible for a human to come up with a good set of rules to describe what constitutes a face in a digital image
- Using machine learning, however, simply presenting a program with a large collection of images of faces is enough for an algorithm to determine what characteristics are needed to identify a face.

Machine learning Types

1. Supervised Learning

- Is machine learning algorithms that automate decision-making processes by generalizing from known examples
- Here , the user provides the algorithm with pairs of inputs and desired outputs, and the algorithm finds a way to produce the desired out put given an input

Example:

  - the user provides the algorithm with a large number of emails (which are the input), together with information about whether any of these emails are spam (which is the desired output). Then given a new email, the algorithm will then produce a prediction as to whether the new email is spam

- Here a "teacher" provides supervision to the algorithms in the form of the desired outputs for each example that they learn from

2. Unsupervised learning

- In unsupervised learning, only the input data is known, and no known output data is given to the algorithm.
- While there are many successful applications of these methods, they are usually harder to understand and evaluate.

Examples of unsupervised learning include:

- Identifying topics in a set of blog posts
  - If you have a large collection of text data, you might want to summarize it and find prevalent themes in it. You might not know beforehand what these topics are, or how many topics there might be. Therefore, there are no known outputs.
- Segmenting customers into groups with similar preferences
  - Given a set of customer records, you might want to identify which customers are similar, and whether there are groups of customers with similar preferences

Supervised Learning

Classification and Regression

- There are two major types of supervised machine learning problems, called classification and regression.

A. Classification

In classification, the goal is to predict a class label, which is a choice from a predefined list of possibilities

Two types of Classification

1. binary classification,

- which is the special case of distinguishing between exactly two classes,
  - trying to answer a yes/no question
  - Classifying emails as either spam or not spam
  - "Is this email spam?"

2. multiclass classification

- which is classification between more than two classes
  - Classifying **iris** flowers in to species setosa, versicolor, or virginica based on the length and width of the **petals** and the length and width of the **sepals**

  - **Reading assignment: go to kaggle.com and explore iris dataset**

B. Regression

- the goal is to predict a continuous number, or a floating-point number in programming terms (or real number in mathematical terms).
- Example:
  - Predicting a person's annual income from their education, their age, and where they live is an example of a regression task
  - predicting the yield of a corn farm given attributes such as previous yields, weather, and number of employees working on the farm
- When predicting income, the predicted value is an amount, and can be any number in a given range.

- An easy way to distinguish between classification and regression tasks is to ask whether there is some kind of **continuity** in the output.
- If there is continuity between possible outcomes, then the problem is a regression problem. Think about predicting annual income
- By contrast, for the task of recognizing the language of a website (which is a classification problem), there is no matter of degree.
  - A website is in one language, or it is in another. There is no continuity between languages, and there is no language that is between English and French

Generalization, Over fitting, and Under fitting

- In supervised learning, we want to build a model on the training data and then be able to make accurate predictions on new, unseen data
- If a model is able to make accurate predictions on unseen data, we say it is able to ***generalize*** from the training set to the test set

Over fitting.

- Is building a model that is too complex for the amount of information we have
- occurs when you fit a model too closely to the particularities of the training set and obtain a model that works well on the training set but is not able to generalize to new data

Table 2-1. Example data about customers

Age	Number of cars owned	Owns house	Number of children	Marital status	Owns a dog	Bought a boat
66	1	yes	2	widowed	no	yes
52	2	yes	3	married	no	yes
22	0	no	0	married	yes	no
25	1	no	1	single	no	no
44	0	no	2	divorced	yes	no
39	1	yes	2	married	yes	no
26	1	no	2	single	no	no
40	3	yes	1	married	yes	no
53	2	yes	2	divorced	no	yes
64	2	yes	3	divorced	no	no
58	2	yes	2	married	yes	yes
33	1	no	1	single	no	no

- "If the customer is older than 45, and has less than 3 children or is not divorced, then they want to buy a boat." When asked how well this rule of his does, our data scientist answers, "It's 100 percent accurate!" And indeed, on the data that is in the table, the rule is perfectly accurate. There are many possible rules we could come up with that would explain perfectly if someone in this dataset wants to buy a boat. No age appears twice in the data, so we could say people who are 66, 52, 53, or 58 years old want to buy a boat, while all others don't. While we can make up many rules that work well on this data, remember that we are not interested in making pre- dictions for this dataset; we already know the answers for these customers. We want to know if new customers are likely to buy a boat.
- We therefore want to find a rule that will work well for new customers, and achieving 100 percent accuracy on the training set does not help us there."
- However, intuitively we expect simple models to generalize better to new data.
- If the rule was "People older than 50 want to buy a boat," and this would explain the behavior of all the customers, we would trust it more than the rule involving children and marital status in addition to age

Under fitting

- Is choosing too simple model
- If your model is too simple then you might not be able to capture all the aspects of and variability in the data
- your model will do badly even on the training set
- Ex “Everybody who owns a house buys a boat”

- The more complex we allow our model to be, the better we will be able to predict on the training data.
- However, if our model becomes too complex, we start focusing too much on each individual data point in our training set, and the model will not generalize well to new data

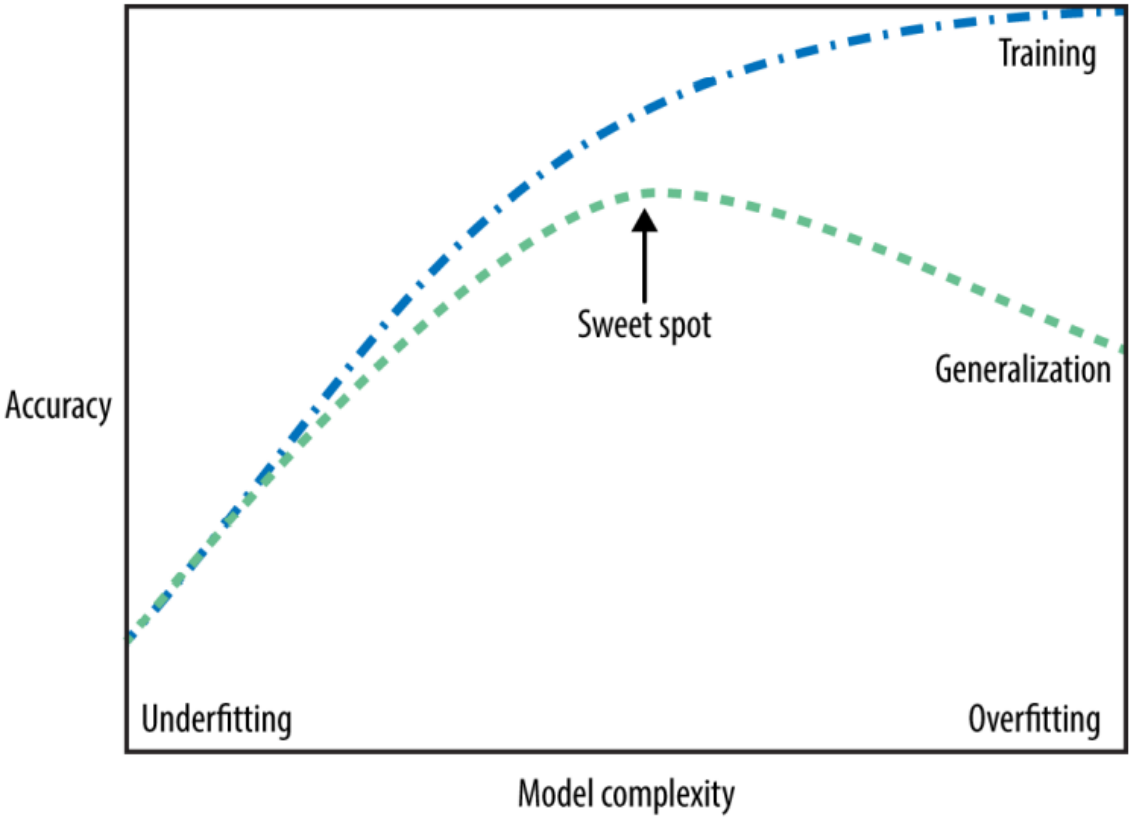


Figure . Trade-off of model complexity against training and test accuracy

Models

- Models approximate reality to let us explore relationships and make predictions
- In machine learning, we build models by training machine learning algorithms with Training Data
- Statistics can be used to determine if a model is useful or believable.
- Finally, we then validate the accuracy of the model using test data

Linear regression

- This algorithm plots a linear relationship between the features(x) and the target(y).
- Feature data values are also called *independent* variables because they are not influenced by anything, they are just the property of that particular dataset.
- Similarly target data values are also called *dependent variables* because they are in some way related to the feature or dependent variables.

price = 135.78 * area + 180616.43

- To understand this better consider the following equation: -

$$y = \theta_0 + \theta_1 x$$
$$where\ \theta_0 \rightarrow intercept$$
$$and\ \theta_1 \rightarrow slope$$

and for more than one feature this can be written as : -

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + ... + \theta_n x_n$$
$$where\ \theta_0 \rightarrow intercept$$
$$and\ \theta_1,\ \theta_2,\ \theta_3,...., \theta_n \rightarrow slope$$

- We know that our data will not all be related in the same linear manner as shown in figure 3.1.
- Based on this, our task in Linear Regression is to find the *best possible relationship (best fit)* for which the error or deviation of the actual target from the target that we get from our relationship *is as small as possible.*
- The mean square error (MSE) reflects the deviation of the actual target from the target we obtained from the relation for each data point.
- The error for the whole dataset is calculated using MSE, which takes *the mean of all the square values* of the deviation of the actual target from the target that we obtained from our relation.

$$MSE = \frac{\sum_{i=1}^{n} (y_{relation} - y_{actual})^2}{N}$$

$$where,\ N \rightarrow total\ number\ of\ data\ points$$

this can be also written as: -

$$MSE = \frac{\sum_{i=1}^{n} (\theta_0 + \theta_1 x - y_{actual})^2}{N}$$

- Now to find the optimal solution we need to minimise the value of MSE, which can be done by partially differentiating the above equation w.r.t intercept variable and slope variable both separately and equating them to 0 as shown in figure 3.2 .
- Below are the differentiated equations.

Differentiating partially w.r.t θ_1 : -

$$\frac{\partial(MSE)}{\partial\theta_1} = \frac{2\sum_{i=1}^{n}(\theta_0 + \theta_1 x - y_{actual})(x)}{N} = 0 \text{ ----- } (1)$$

and marking it as eq^n 1

Now, differentiating partially w.r.t θ_0: -

$$\frac{\partial(MSE)}{\partial\theta_0} = \frac{2\sum_{i=1}^{n}(\theta_0 + \theta_1 x - y_{actual})}{N} = 0 \text{ ----- } (2)$$

and marking it as eq^n 2

Calculating MSE

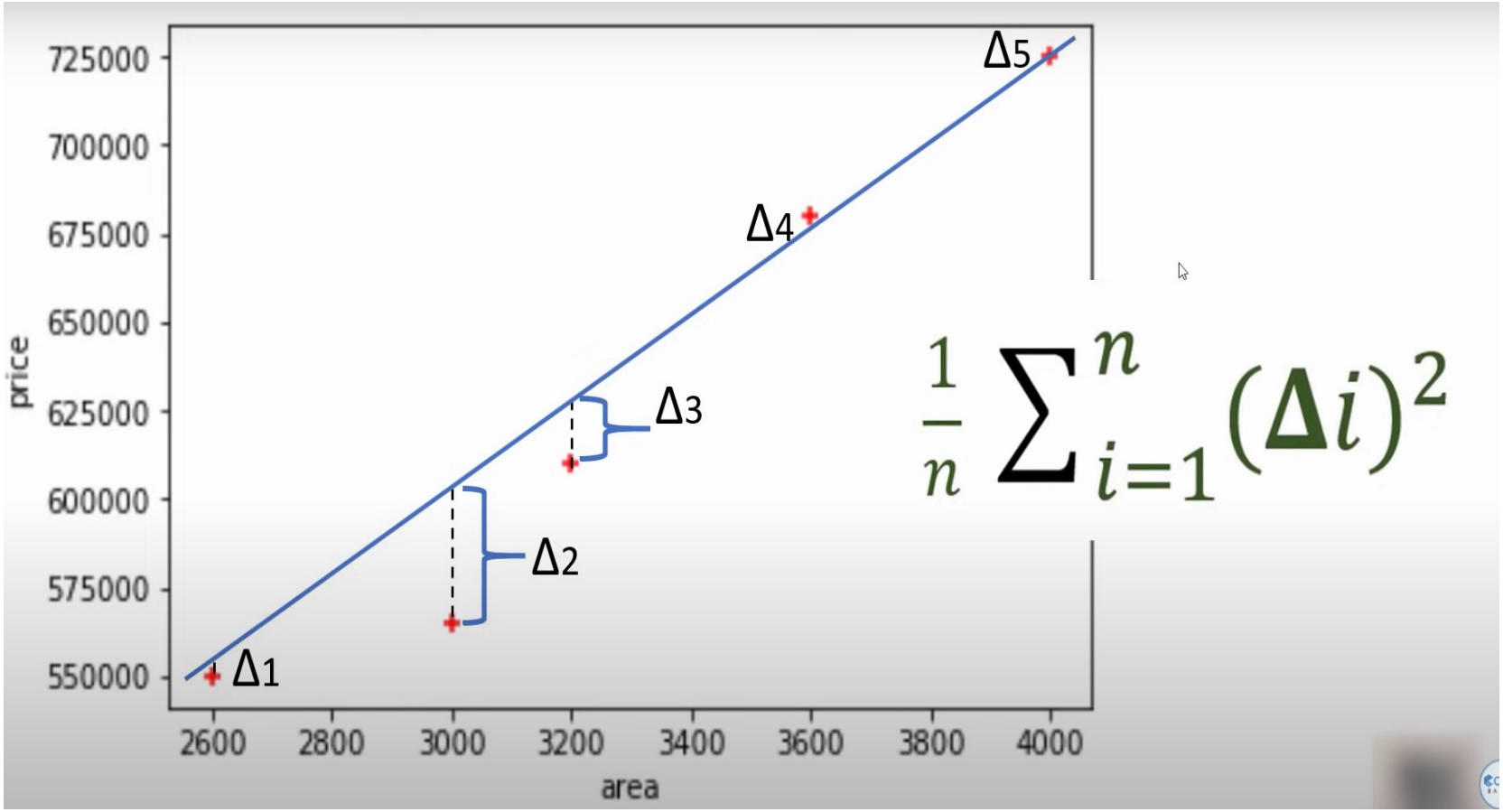


Figure 3.1

Note: Linear models(Linear regression) are just about fitting the best line to the data points

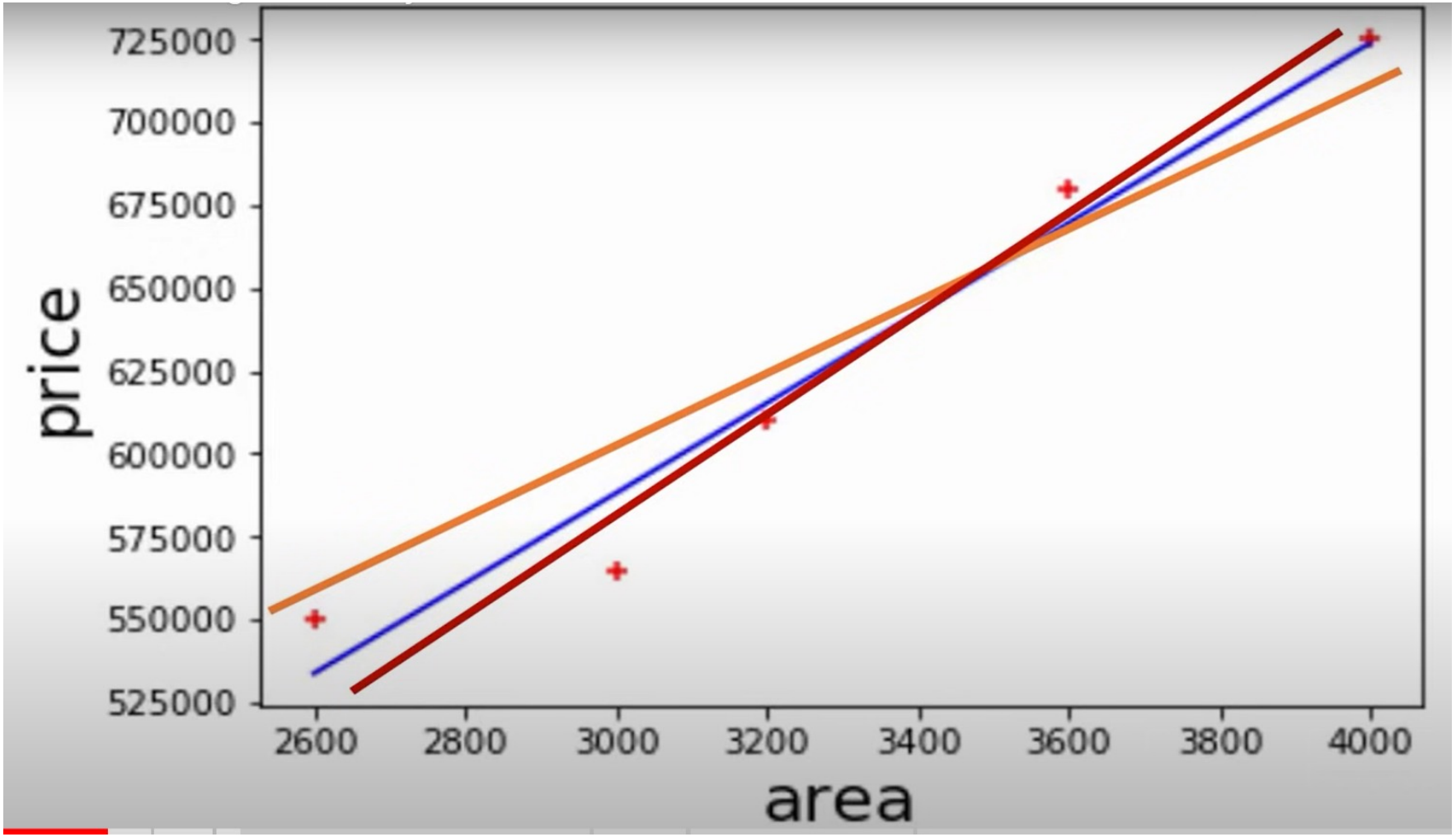


Figure 3.2

Model Accuracy measures

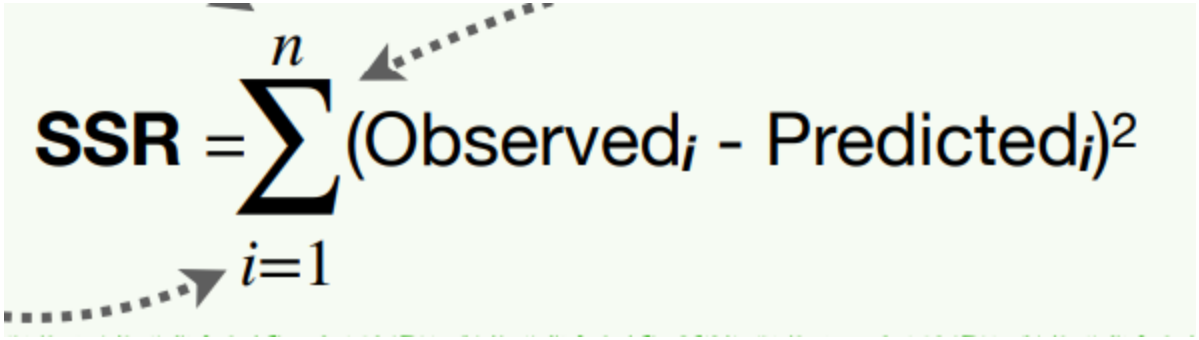
- Now let's talk about how statistics can quantify the quality of a model. There are various metrics used.

1. The Sum of the Squared Residuals

- As the name implies, we start by calculating Residuals, the differences between the Observed values and the values Predicted by the model
- in general, the smaller the Residuals, the better the model fits the data

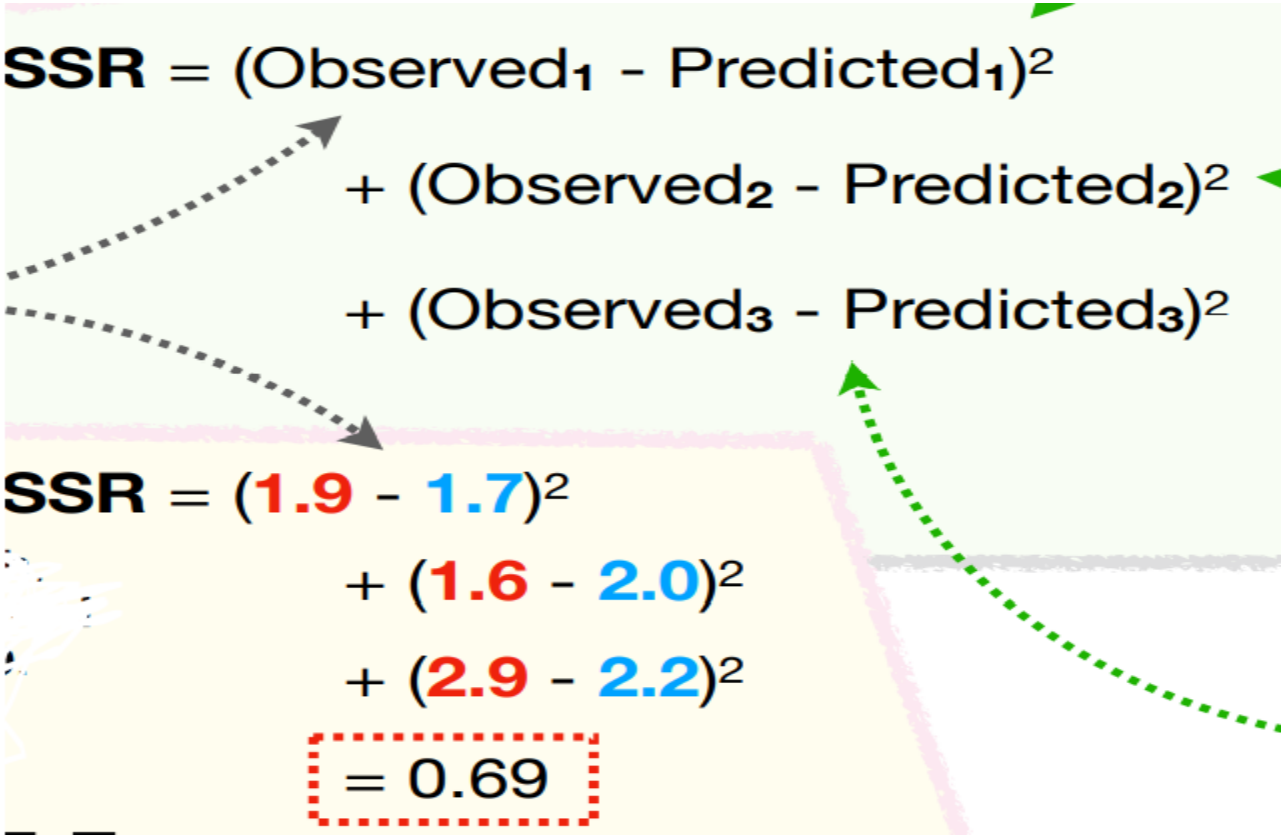
2. The Sum of Squared Residuals (SSR)

- The sum of all observations of the squared difference between the observed and predicted values.



n = the number of Observations
i = the index for each Observation. For example, i = 1 refers to the first Observation

- For instance if you have 1.9, 1.6 and 2.9 observations and the predicted values by the model for these values are 1.7, 2, and 2.2 respectively then the model accuracy would be using SSR:



3. Mean squared error(MSE)

- Sum of the Squared Residuals (SSR), is not super easy to interpret because it depends, in part, on how much data you have
- For example, if we start with a simple dataset with 3 points, the Residuals are, from left to right, 1, -3, and 2, and the SSR = 14
- Now, if we have a second dataset that includes 2 more data points added to the first one, and the Residuals are -2 and 2, then the SSR increases to 22
- However, the increase in the SSR from 14 to 22 does not suggest that the second model, fit to the second, larger dataset, is worse than the first. It only tells us that the model with more data has more Residuals

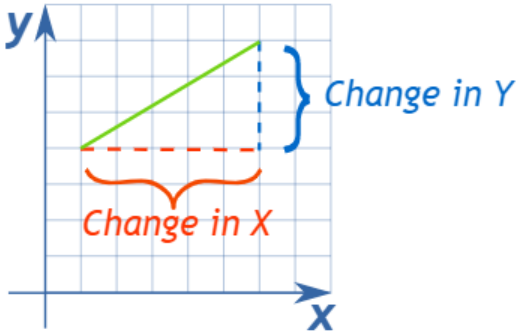
- A Solution: One way to compare the two models that may be fit to different-sized datasets is to calculate the Mean Squared Error (MSE), which is simply the average of the SSR

$$\text{Mean Squared Error (MSE)} = \frac{\text{SSR}}{n} = \sum_{i=1}^{n} \frac{(\text{Observed}_i - \text{Predicted}_i)^2}{n}$$

Introduction to Derivatives

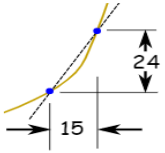
It is all about slope!

$$\text{Slope} = \frac{\text{Change in Y}}{\text{Change in X}}$$



Introduction to Derivatives

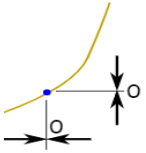
We can find an **average** slope between two points.



average slope = $\frac{24}{15}$

But how do we find the slope **at a point**?

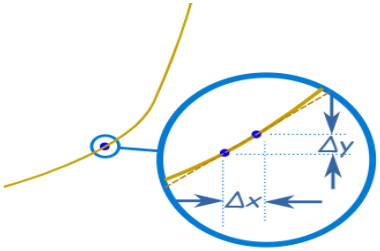
There is nothing to measure!



slope = $\frac{0}{0}$ = ???

But with derivatives we use a small difference ...

... then have it **shrink towards zero**.

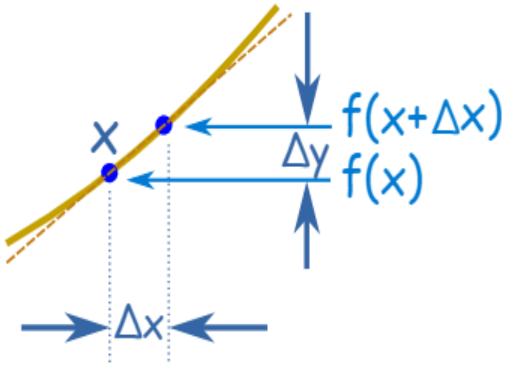


Let us Find a Derivative!

To find the derivative of a function $y = f(x)$ we use the slope formula:

$$\text{Slope} = \frac{\text{Change in Y}}{\text{Change in X}} = \frac{\Delta y}{\Delta x}$$

And (from the diagram) we see that:



x changes from x to $x+\Delta x$

y changes from $f(x)$ to $f(x+\Delta x)$

Now follow these steps:

- Fill in this slope formula: $\frac{\Delta y}{\Delta x} = \frac{f(x+\Delta x) - f(x)}{\Delta x}$
- Simplify it as best we can
- Then make **Δx** shrink towards zero.

Example: the function $\mathbf{f(x) = x^2}$

We know $\mathbf{f(x) = x^2}$, and we can calculate $\mathbf{f(x+\Delta x)}$:

Start with: $\mathbf{f(x+\Delta x) = (x+\Delta x)^2}$

Expand $(x + \Delta x)^2$: $\mathbf{f(x+\Delta x) = x^2 + 2x\,\Delta x + (\Delta x)^2}$

The slope formula is:	$\frac{f(x+\Delta x) - f(x)}{\Delta x}$
Put in $\mathbf{f(x+\Delta x)}$ and $\mathbf{f(x)}$:	$\frac{x^2 + 2x\,\Delta x + (\Delta x)^2 - x^2}{\Delta x}$
Simplify (x^2 and $-x^2$ cancel):	$\frac{2x\,\Delta x + (\Delta x)^2}{\Delta x}$
Simplify more (divide through by Δx):	$= 2x + \Delta x$
Then, **as Δx heads towards 0** we get:	$= 2x$

Result: the derivative of $\mathbf{x^2}$ is $\mathbf{2x}$

In other words, the slope at x is **2x**

We write **dx** instead of **"Δx heads towards 0"**.

And "the derivative of" is commonly written $\frac{d}{dx}$ like this:

$$\frac{d}{dx}x^2 = 2x$$

"The derivative of $\boldsymbol{x^2}$ *equals* $\boldsymbol{2x}$*"*

or simply *"d dx of* $\boldsymbol{x^2}$ *equals* $\boldsymbol{2x}$*"*

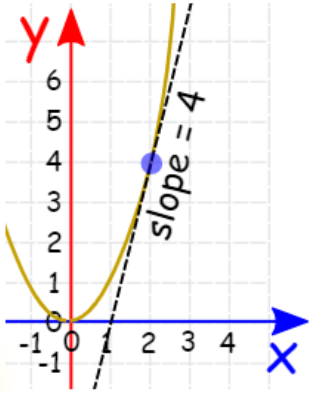
Introduction to Derivatives

So what does $\frac{d}{dx}x^2 = 2x$ mean?

It means that, for the function x^2, the slope or "rate of change" at any point is **2x**.

So when **x=2** the slope is **2x = 4**, as shown here:

Or when **x=5** the slope is **2x = 10**, and so on.



Note: f′(x) can also be used for "the derivative of":

$$f'(x) = 2x$$

"The derivative of f(x) equals 2x"

or simply *"f-dash of x equals 2x"*

Example: What is $\frac{d}{dx}x^3$?

We know **$f(x) = x^3$**, and can calculate **$f(x+\Delta x)$** :

Start with: **$f(x+\Delta x) = (x+\Delta x)^3$**

Expand $(x + \Delta x)^3$: **$f(x+\Delta x) = x^3 + 3x^2\,\Delta x + 3x\,(\Delta x)^2 + (\Delta x)^3$**

The slope formula:	$\frac{f(x+\Delta x) - f(x)}{\Delta x}$
Put in **$f(x+\Delta x)$** and **$f(x)$**:	$\frac{x^3 + 3x^2\,\Delta x + 3x\,(\Delta x)^2 + (\Delta x)^3 - x^3}{\Delta x}$
Simplify (x^3 and $-x^3$ cancel):	$\frac{3x^2\,\Delta x + 3x\,(\Delta x)^2 + (\Delta x)^3}{\Delta x}$
Simplify more (divide through by Δx):	$3x^2 + 3x\,\Delta x + (\Delta x)^2$
Then, **as Δx heads towards 0** we get:	$3x^2$

Result: the derivative of **x^3** is **$3x^2$**

Derivative Rules

1. Power Rule of Differentiation

$$d/dx(x^n) = nx^{n-1}$$

Example: Find the derivative of x^5

Solution: As per the power rule, we know;

$$d/dx(x^n) = nx^{n-1}$$

Hence, $d/dx(x^5) = 5x^{5-1} = 5x^4$

2. Sum Rule of Differentiation

If $f(x)=u(x)\pm v(x)$, then;

$$f'(x)=u'(x)\pm v'(x)$$

Derivative Rules

Example 1: $f(x) = x + x^3$

Solution: By applying sum rule of derivative here, we have:

$$f'(x) = u'(x) + v'(x)$$

Now, differentiating the given function, we get;

$$f'(x) = d/dx(x + x^3)$$

$$f'(x) = d/dx(x) + d/dx(x^3)$$

$$f'(x) = 1 + 3x^2$$

Exercise:**Find the derivative of the function $f(x) = 6x^2 – 4x$.**

3. Product Rule of Differentiation

If $f(x) = u(x) \times v(x)$, then:

$f'(x) = u'(x) \times v(x) + u(x) \times v'(x)$

Example: Find the derivative of $x^2(x+3)$.

$f'(x) = u'(x) \times v(x) + u(x) \times v'(x)$

Here,

$u(x) = x^2$ and $v(x) = x+3$

Therefore, on differentiating the given function, we get;

$f'(x) = d/dx[x^2(x+3)]$

$f'(x) = d/dx(x^2)(x+3)+x^2 d/dx(x+3)$

$f'(x) = 2x(x+3)+x^2(1)$

$f'(x) = 2x^2+6x+x^2$

$f'(x) = 3x^2+6x$

$f'(x) = 3x(x+2)$

4. Quotient Rule of Differentiation

- If f(x) is a function, which is equal to ratio of two functions u(x) and v(x) such that;

$$f(x) = u(x)/v(x)$$

- Then, as per the quotient rule, the derivative of f(x) is given by;

$$\mathbf{f'(x) = \frac{u'(x)\times v(x) - u(x)\times v'(x)}{(v(x))^2}}$$

Example: Differentiate $f(x)=(x+2)^3/\sqrt{x}$

Solution: Given,

$f(x)=(x+2)^3/\sqrt{x}$

$= (x+2)(x^2+4x+4)/\sqrt{x}$

$= [x^3+6x^2+12x+8]/x^{1/2}$

$= x^{-1/2}(x^3+6x^2+12x+8)$

$= x^{5/2}+6x^{3/2}+12x^{1/2}+8x^{-½}$

Now, differentiating the given equation, we get;

$f'(x) = 5/2x^{3/2} + 6(3/2x^{1/2})+12(1/2x^{-1/2})+8(-1/2x^{-3/2})$

$= 5/2x^{3/2} + 9x^{1/2} + 6x^{-½} - 4x^{-3/2}$

5. Chain Rule of Differentiation

If a function y = f(x) = g(u) and if u = h(x), then the chain rule for differentiation is defined as;

$$dy/dx = (dy/du) \times (du/dx)$$

Example 1:

Differentiate $f(x) = (x^4 - 1)^{50}$

Solution:

Given,

$f(x) = (x^4 - 1)^{50}$

Let $g(x) = x^4 - 1$ and $n = 50$

$u(t) = t^{50}$

Thus, $t = g(x) = x^4 - 1$

$f(x) = u(g(x))$

According to chain rule,

$df/dx = (du/dt) \times (dt/dx)$

Here,

$du/dt = d/dt\ (t50) = 50t^{49}$

$dt/dx = d/dx\ g(x)$

$= d/dx\ (x^4 - 1)$

$= 4x^3$

Thus, $df/dx = 50t^{49} \times (4x^3)$

$= 50(x4 - 1)49 \times (4x3)$

$= 200\ x3(x4 - 1)49$

6. Partial derivative

Given function: $f(x,y) = 3x + 4y$

To find $\partial f/\partial x$, keep y as constant and differentiate the function:

Therefore, $\partial f/\partial x = 3$

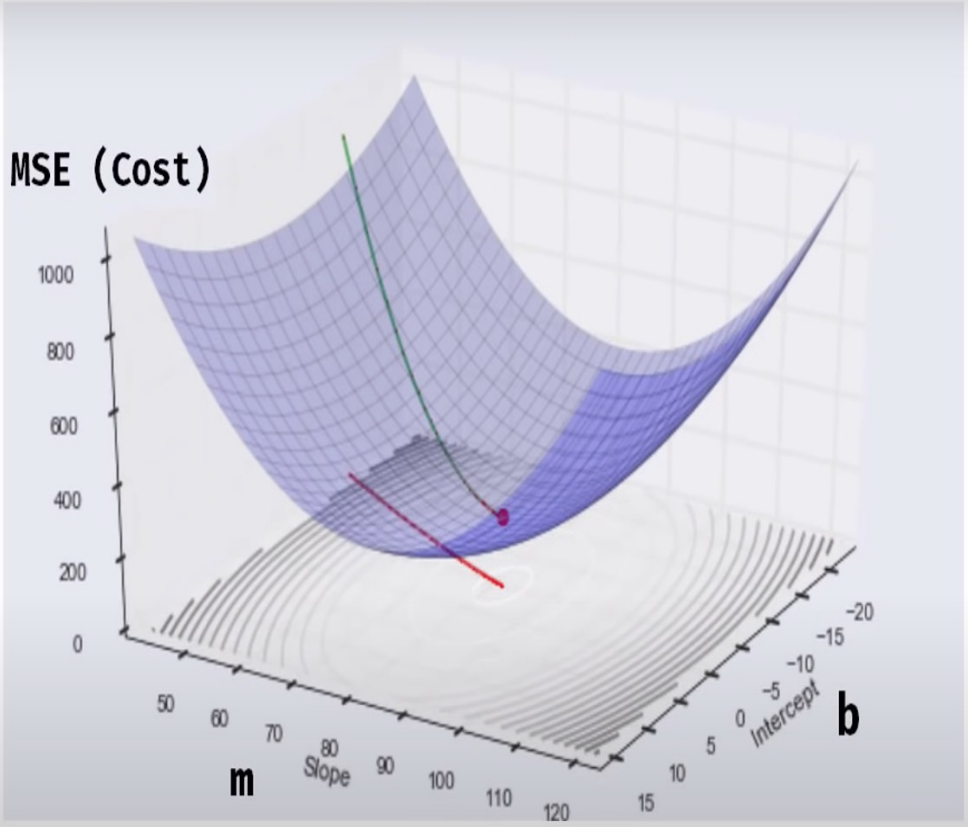
Similarly, to find $\partial f/\partial y$, keep x as constant and differentiate the function:

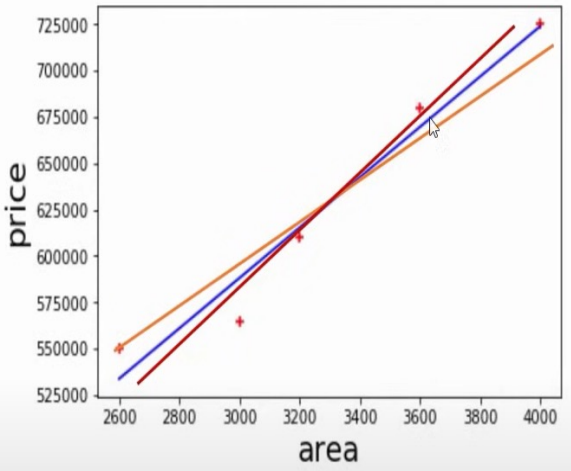
Therefore, $\partial f/\partial y = 4$

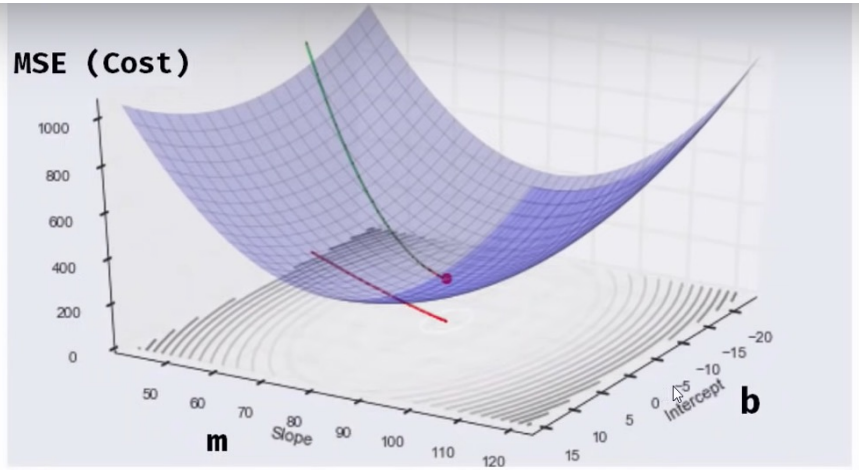
Gradient Descent

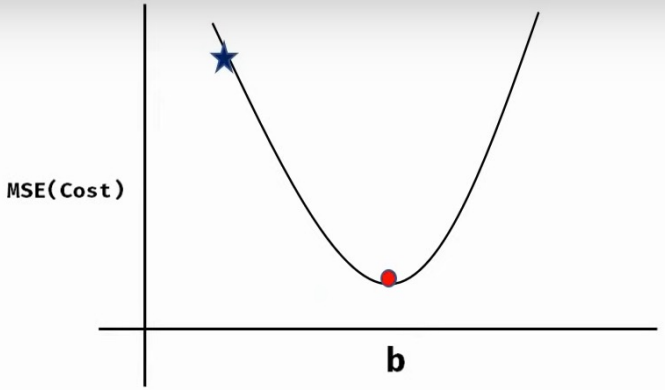
- A major part of machine learning is optimizing a model's fit to the data.
- Sometimes this can be done with an analytical solution, but it's not always possible
- Gradient Descent is an *iterative solution* that incrementally steps toward an optimal solution and is used in a very wide variety of situations.
- 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

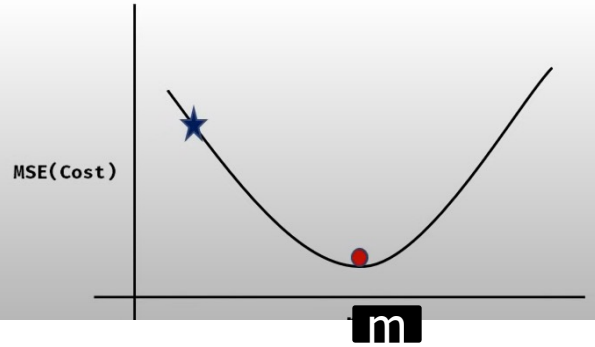
Visualizing gradient descent



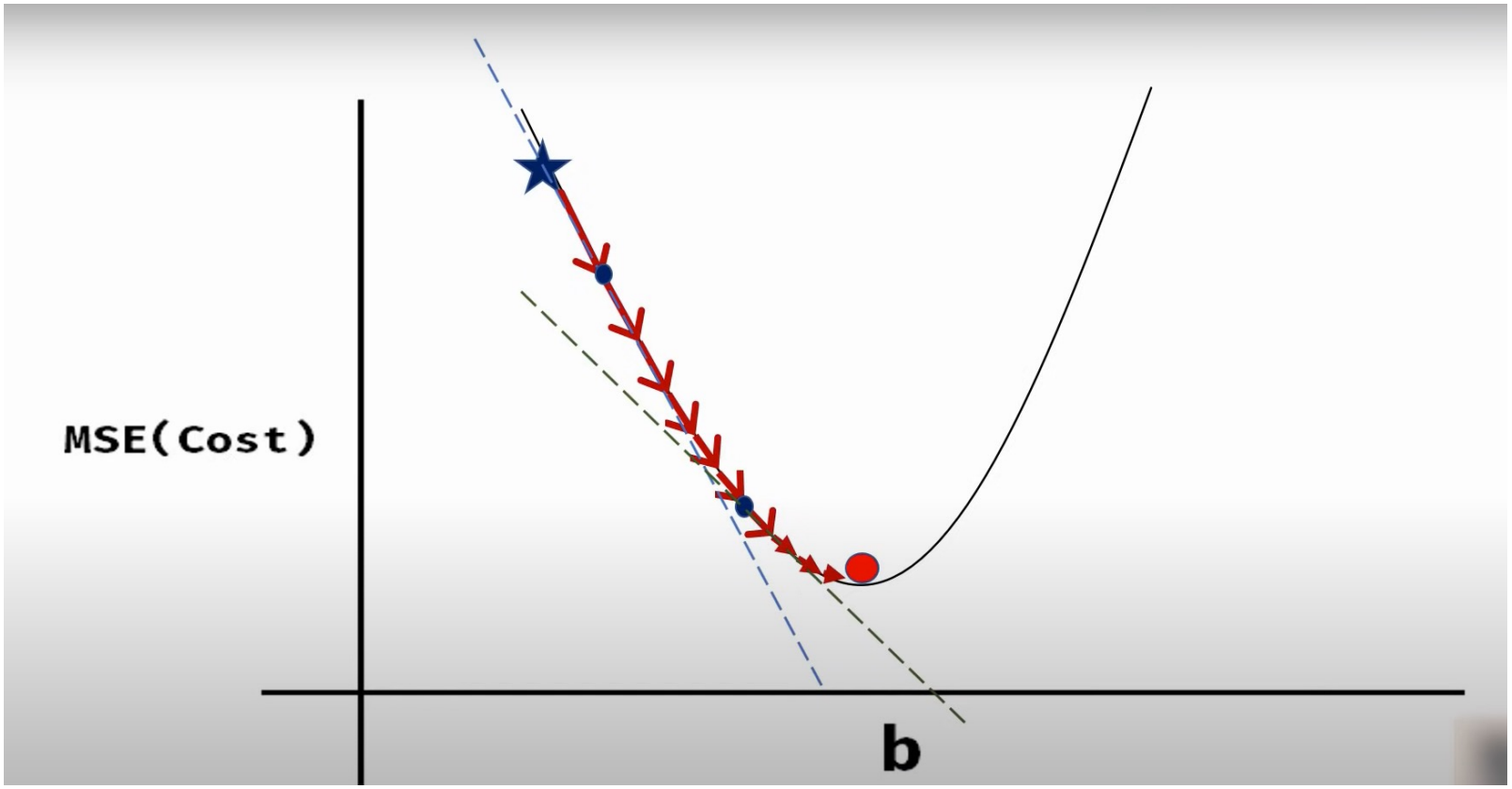



MSE (Cost)
1000
800
600
400
200
0
50
60
70
80
90
100
110
120
m
Slope
15
10
5
0
−5
−10
−15
−20
Intercept
b

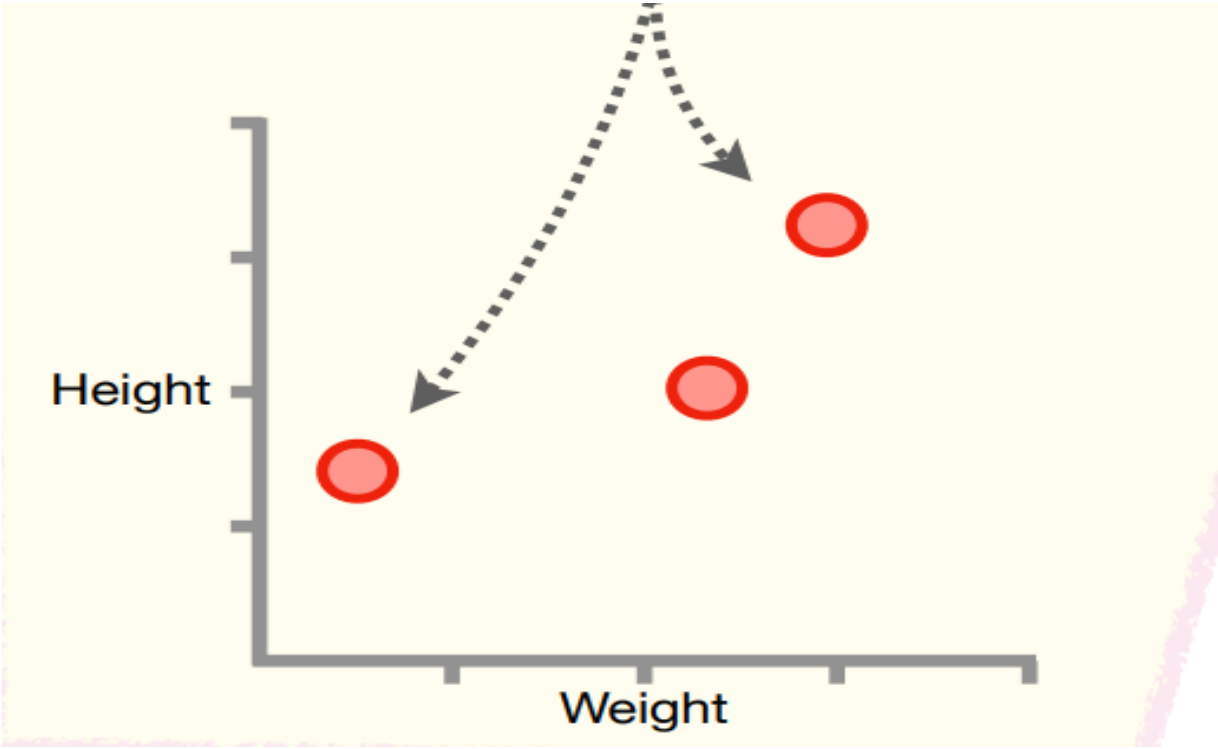

MSE(Cost)
b


MSE(Cost)
m

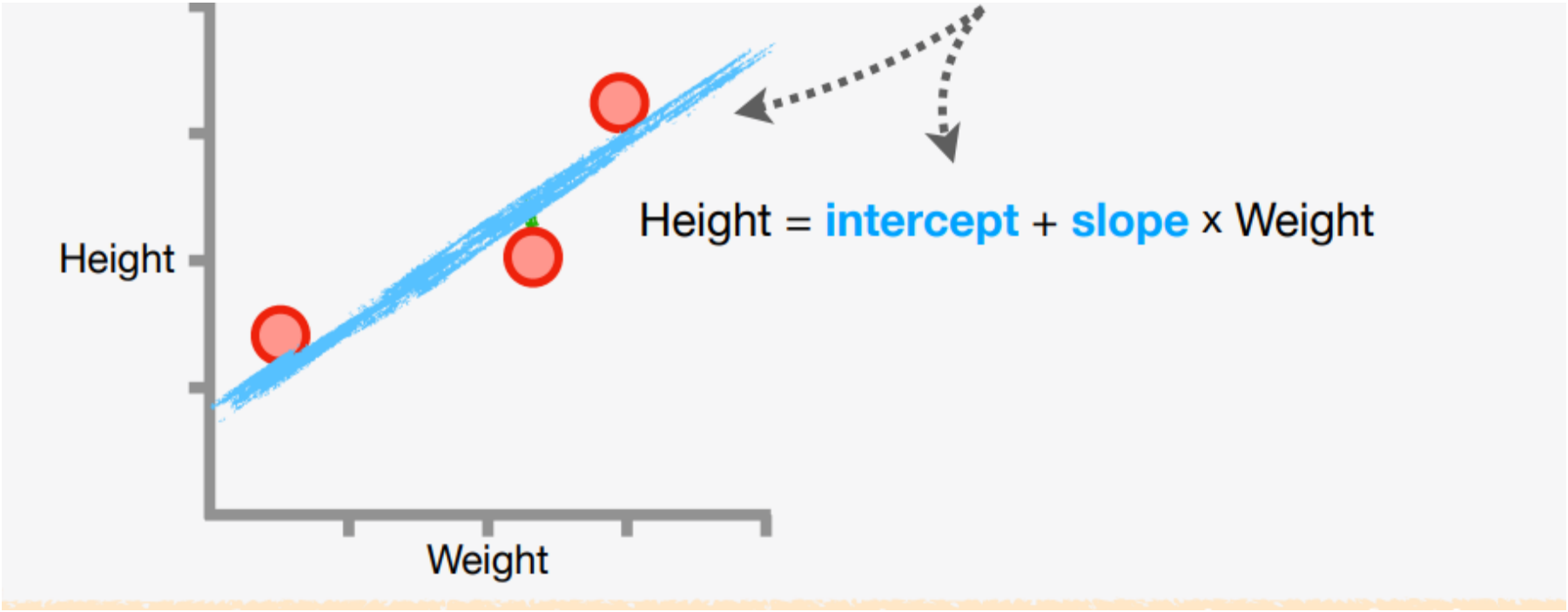
How to make derivation using tangent line and taking baby steps gradually



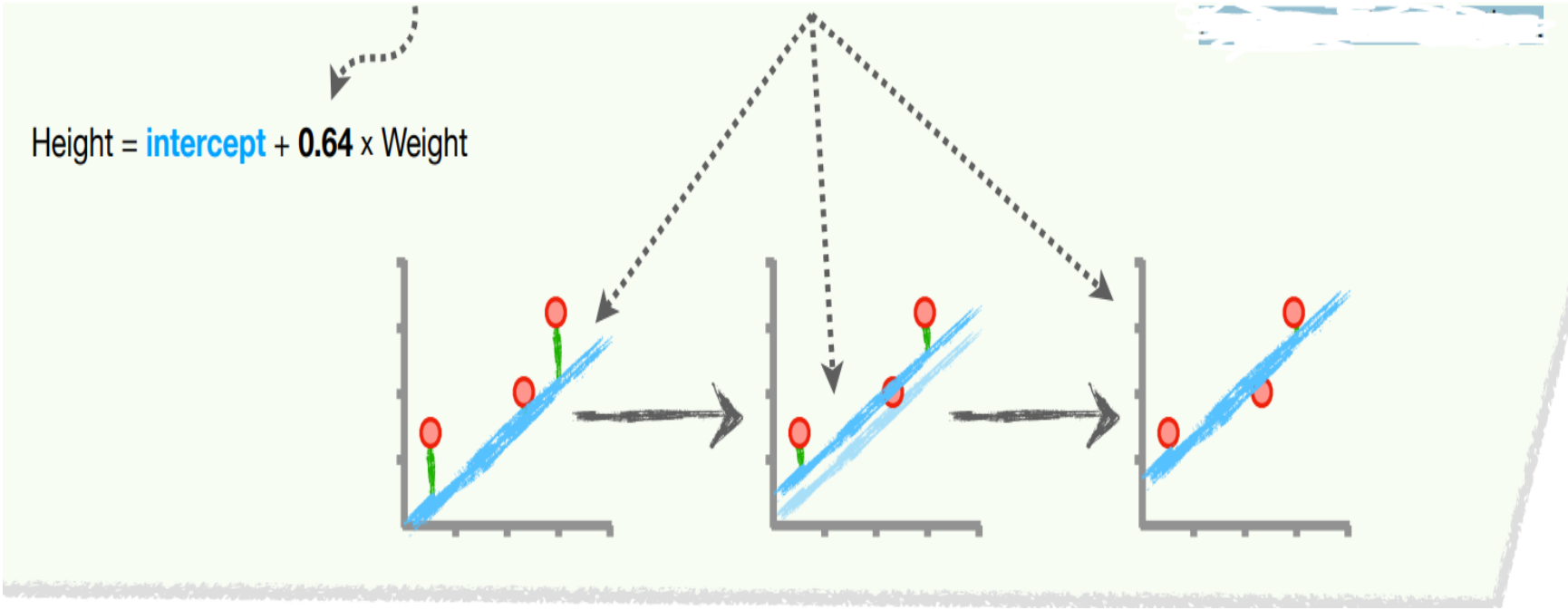
- Let's show how Gradient Descent fits a line to these Height and Weight measurements(*That is to find a linear regression model(formula) that fits how height changes with respect to weight*



- 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…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.



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

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

- …and the Observed Heights are the values we originally measured… …and the Predicted Heights come from the equation for the line

- 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
- Now, to calculate the SSR, we first plug the value for the y-axis intercept, 0, into the equation

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

$$SSR = (\text{Observed Height}_1 - (\textbf{0} + 0.64 \times \text{Weight}_1))^2 + (\text{Observed Height}_2 - (\textbf{0} + 0.64 \times \text{Weight}_2))^2 + (\text{Observed Height}_3 - (\textbf{0} + 0.64 \times \text{Weight}_3))^2$$

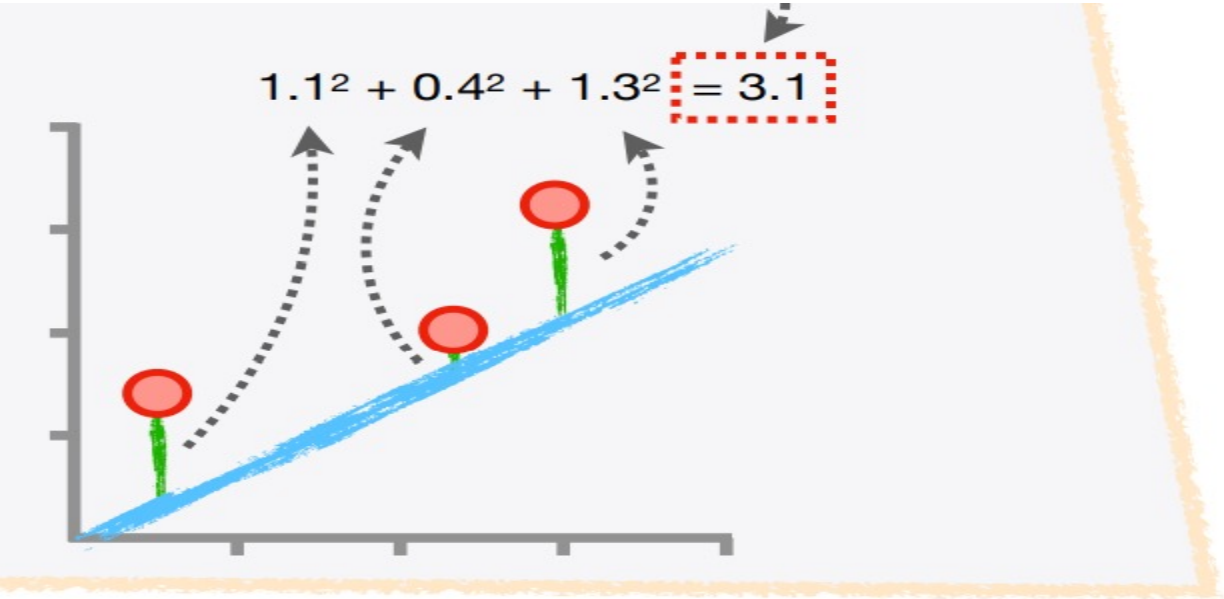
- then we plug in the Observed values for Height and Weight for each data point

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

$$SSR = (1.4 - (0 + 0.64 \times 0.5))^2 + (1.9 - (0 + 0.64 \times 2.3))^2 + (3.2 - (0 + 0.64 \times 2.9))^2$$

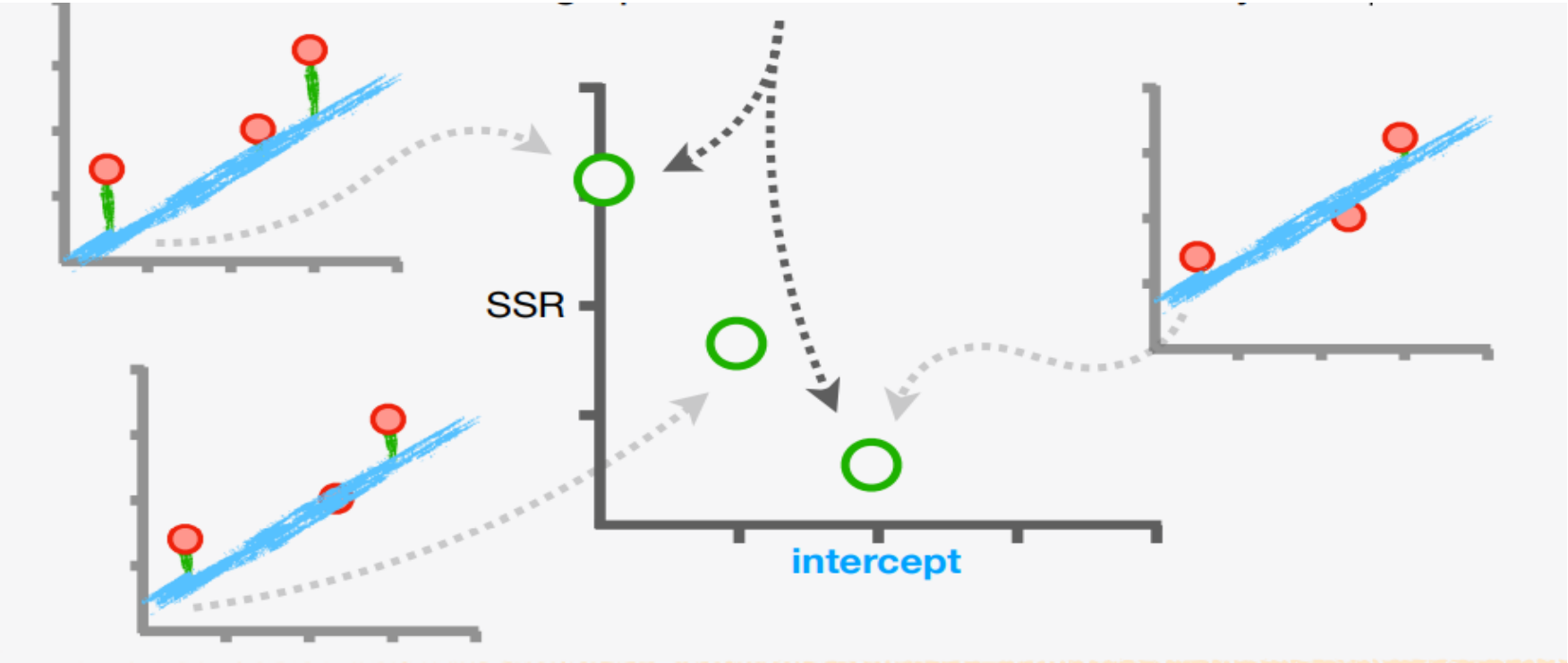
- Lastly, we just do the math. The SSR for when the y-axis intercept is set to 0 is 3.1

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

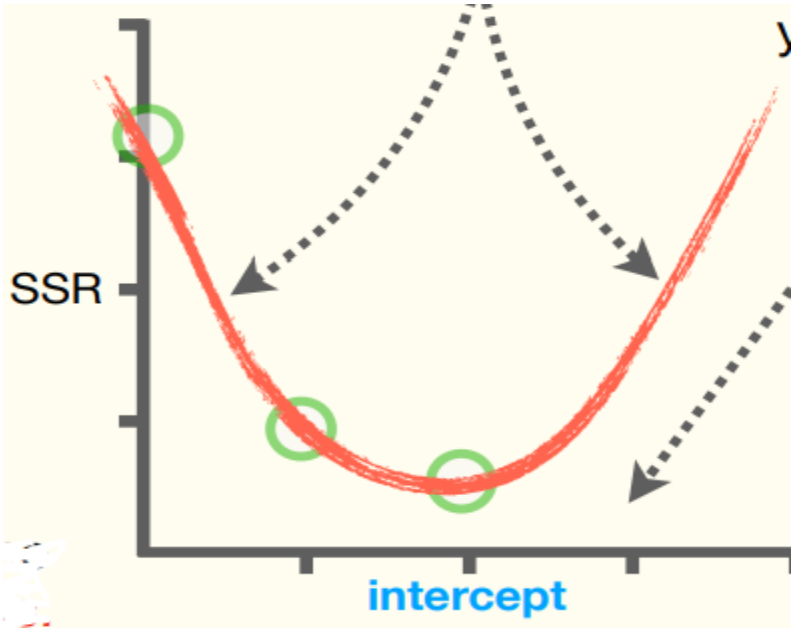


- Now, because the goal is to minimize the SSR, it's a type of Loss or Cost Function.

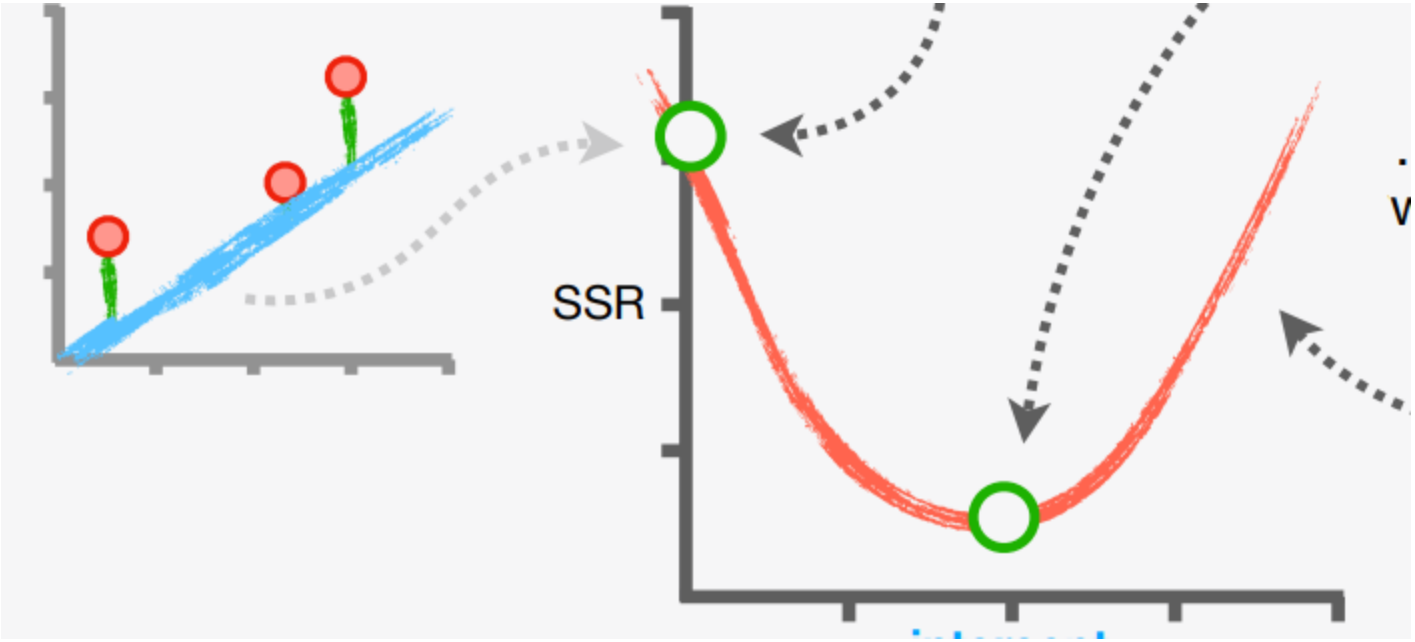
- In Gradient Descent, we minimize the Loss or Cost Function by taking steps away from the initial guess toward the optimal value.
- In this case, we see that as we increase the intercept, the x-axis of the central graph, we decrease the SSR, the y-axis



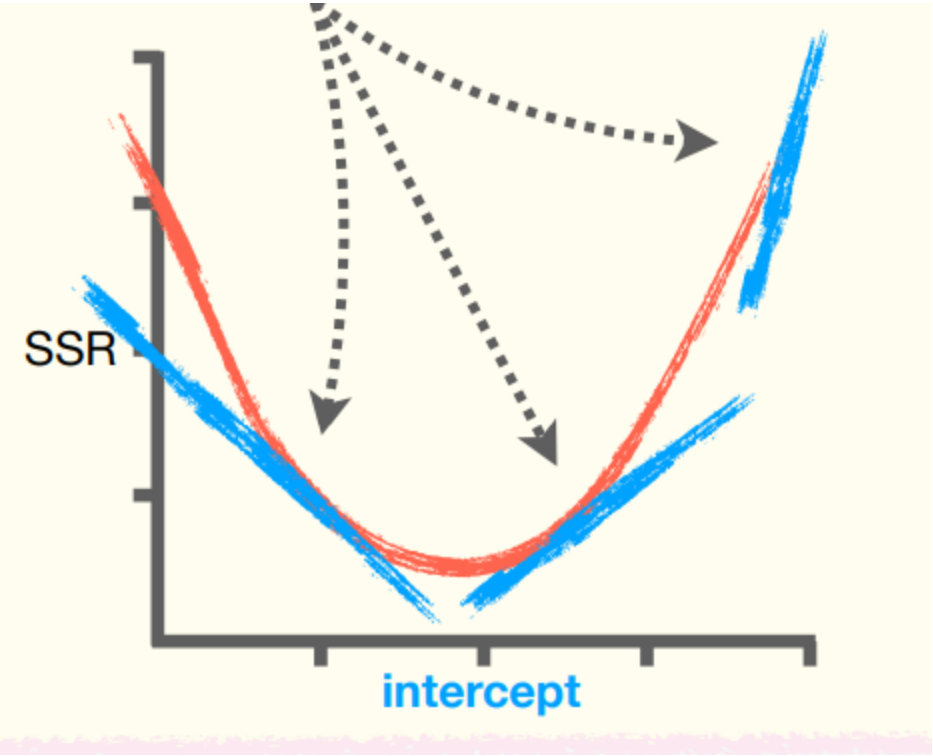
- However, rather than just randomly trying a bunch of values for the y-axis intercept and plotting the resulting SSR on a graph, we can plot the SSR as a function of the y-axis intercept. In other words, the above equation for the SSR corresponds to this curve on a graph that has the SSR on the y-axis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



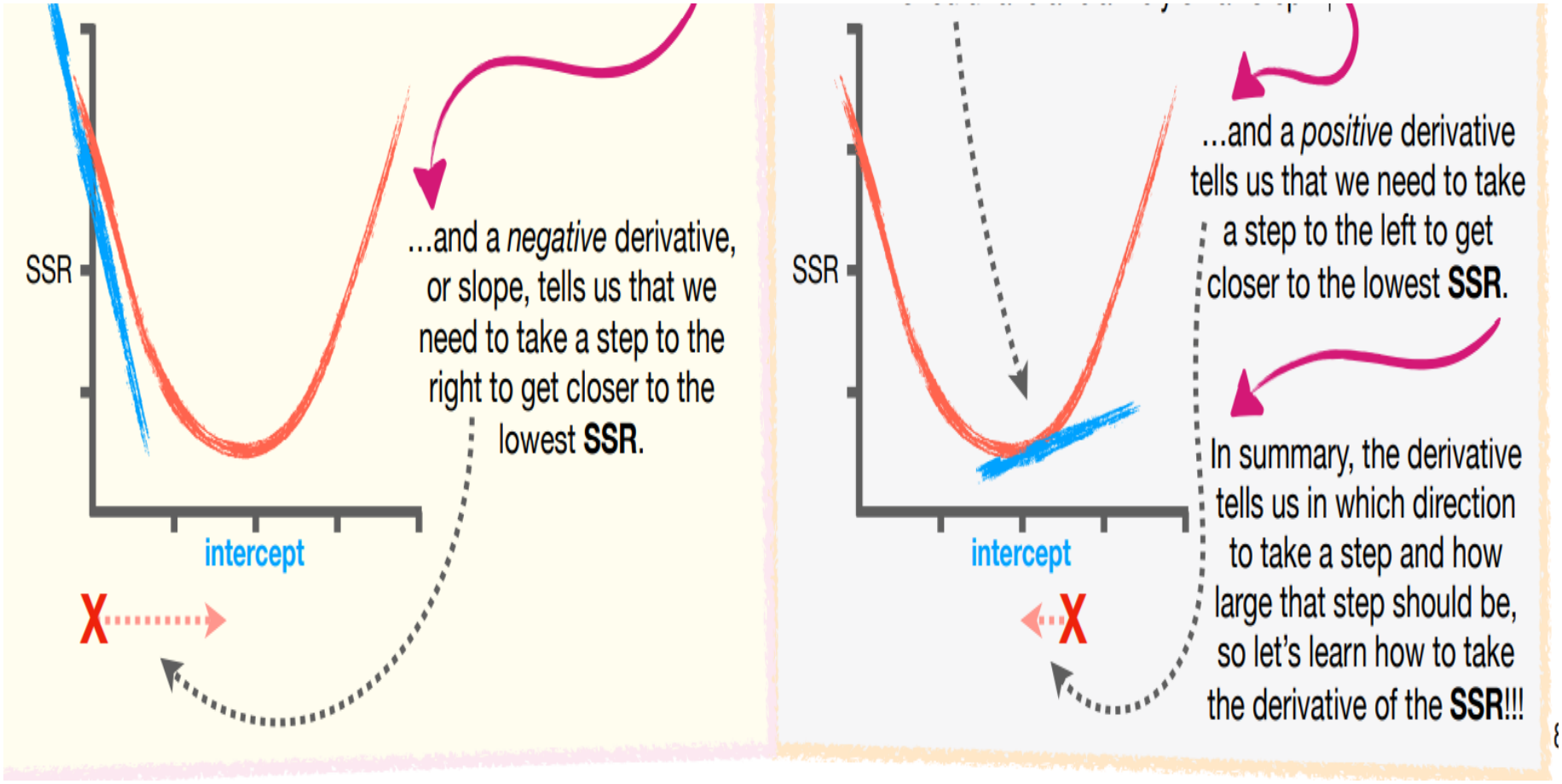
- Now, when we started with the y-axis intercept = 0, we got this SSR …so how do we take steps toward this y-axis intercept that gives us the lowest SSR… …and how do we know when to stop or if we've gone too far?



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



- 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*
- 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*…


SSR
intercept
...and a *negative* derivative, or slope, tells us that we need to take a step to the right to get closer to the lowest **SSR**.
SSR
intercept
...and a *positive* derivative tells us that we need to take a step to the left to get closer to the lowest **SSR**.
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**!!!

How Gradient Descent uses chain rule Derivative to find the lowest SSR to fit best line

SSR = (Height - (intercept + 0.64 x Weight)

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

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

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

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

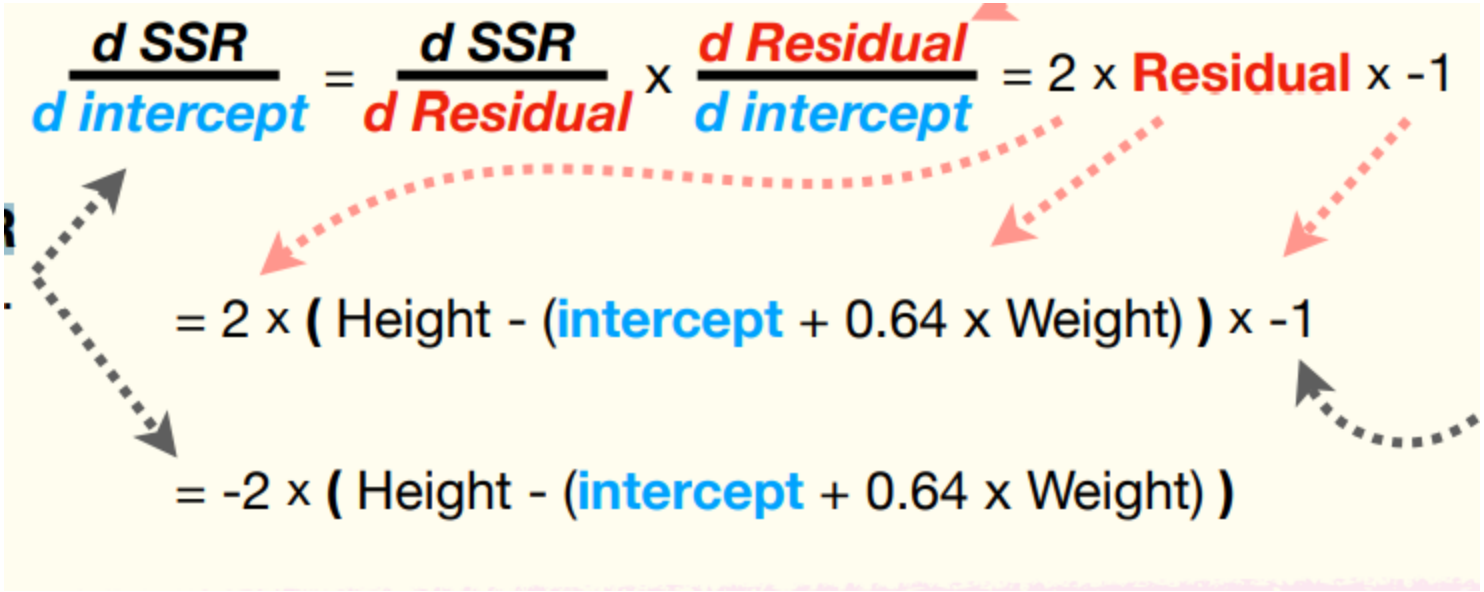
$$\frac{d\,Residual}{d\,intercept} = \frac{d}{d\,intercept}\,\text{Height} - (\text{intercept} + 0.64 \times \text{Weight})$$

$$= \frac{d}{d\,intercept}\,\text{Height} - \text{intercept} - 0.64 \times \text{Weight}$$

$=0 - 1 - 0 = -1$

(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)

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



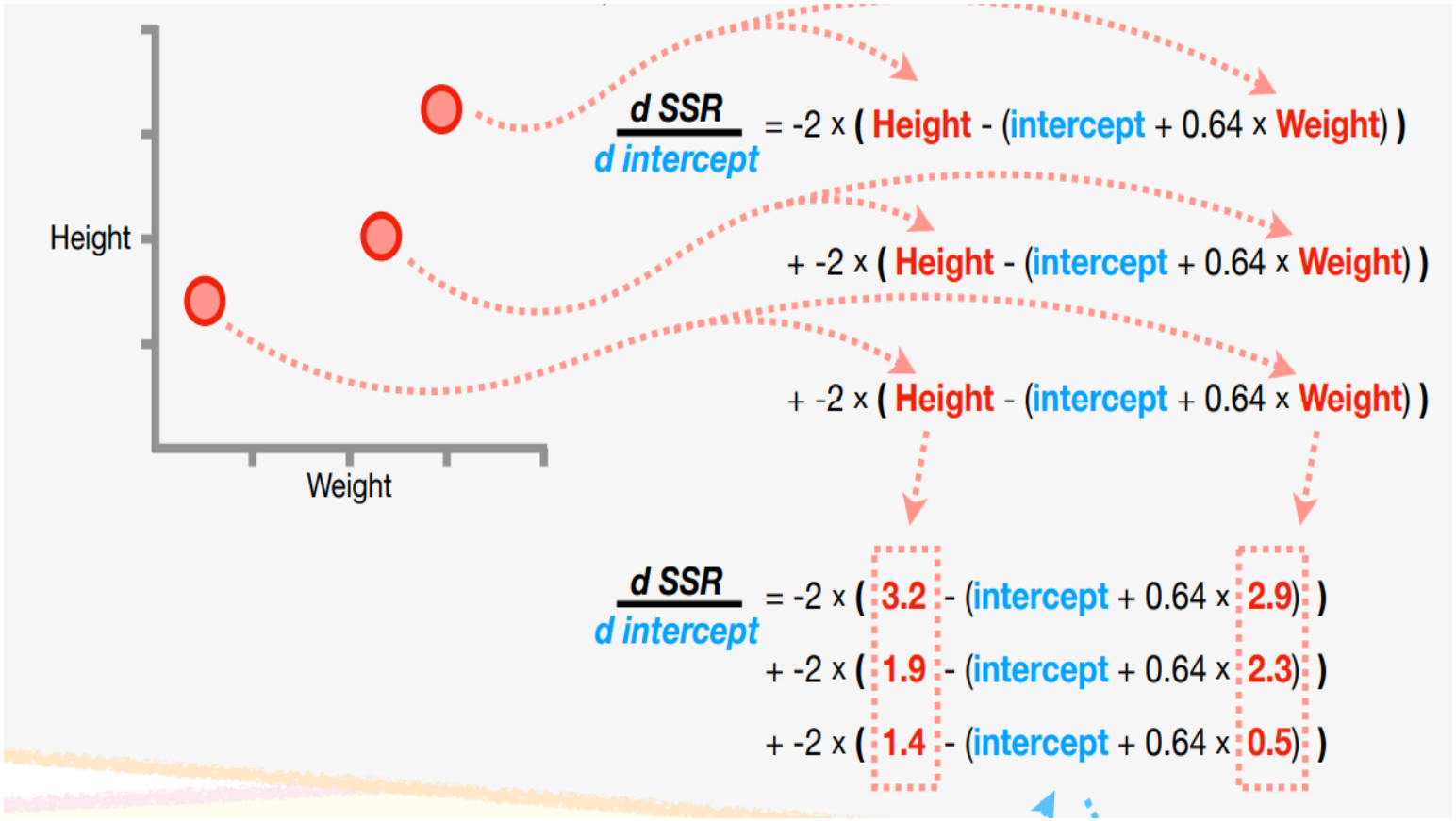
- So far, we've calculated the derivative of the SSR for a single observation.
- However, we have three observations in the dataset, so the SSR and its derivative both have three terms

- The derivative of SSR for the three data points

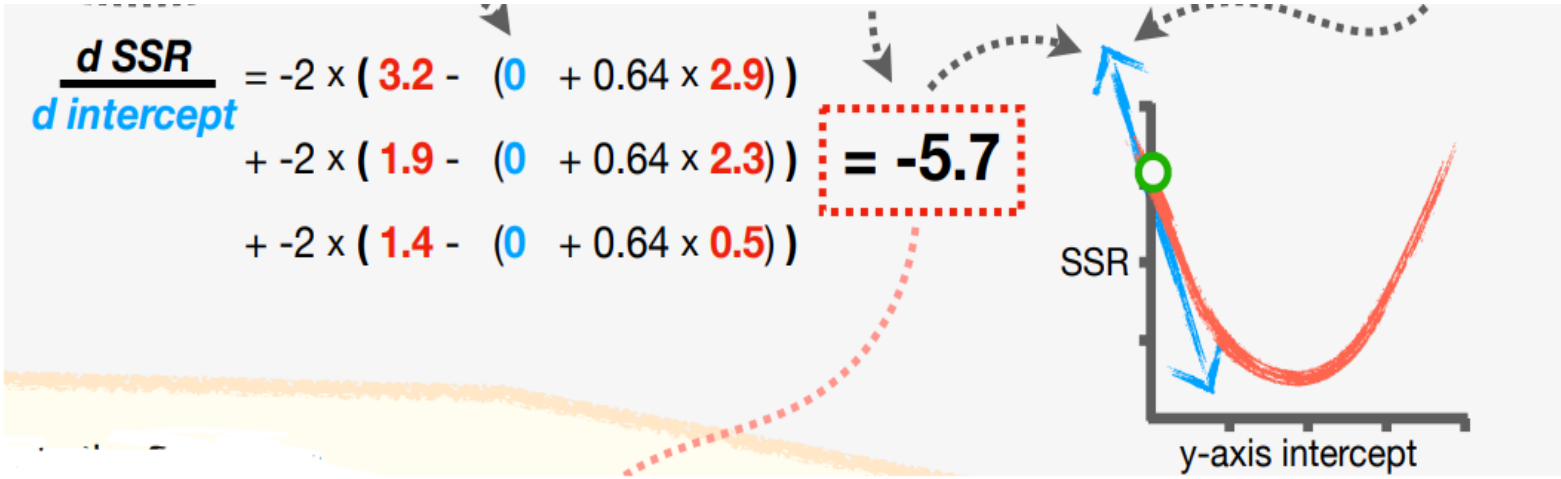
$$SSR = (\text{Height} - (\text{intercept} + 0.64 \times \text{Weight}))^2 + (\text{Height} - (\text{intercept} + 0.64 \times \text{Weight}))^2 + (\text{Height} - (\text{intercept} + 0.64 \times \text{Weight}))^2$$

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

- Step5: 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



- Step 6: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 and Now evaluate the derivative at the current value for the intercept.



- Step 7: Now calculate the Step Size with the following equation

  Step Size = Derivative x Learning Rate

  Step Size = -5.7 x 0.1

  Step Size = -0.57

- **Note**:
  - 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
- **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.

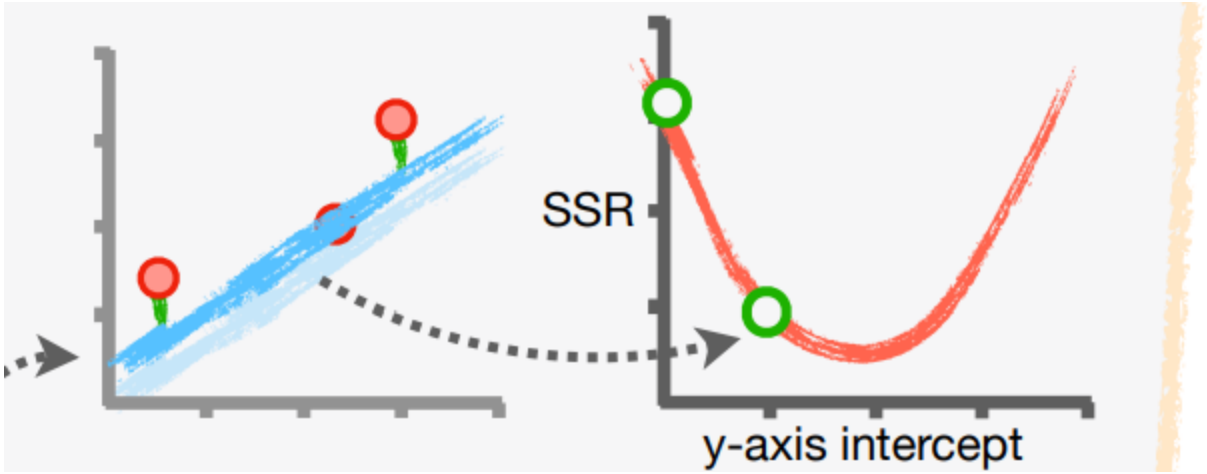
Step 8:Take a step from the current intercept to get closer to the optimal value with the following equation

- Remember, in this case, the current intercept is 0

  New intercept = Current intercept - Step Size

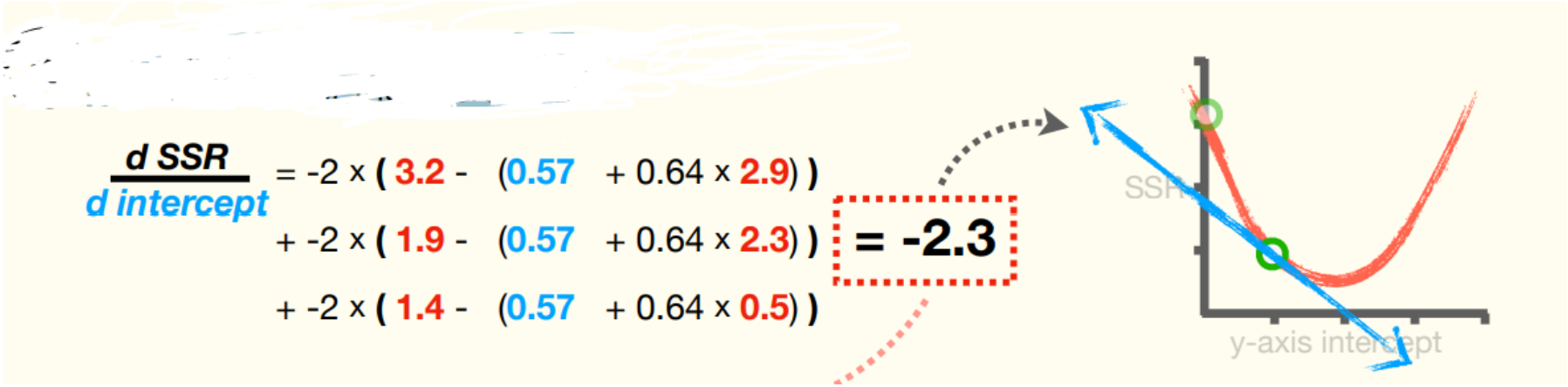
  = 0 - (-0.57) = 0.57

- The new intercept, 0.57, moves the line up a little closer to the data and it results in a lower SSR as shown below.



- Step 9: 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…



b. Calculate the Step Size

Step Size = Derivative x Learning Rate

Step Size = -2.3 x 0.1

Step Size = -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.*

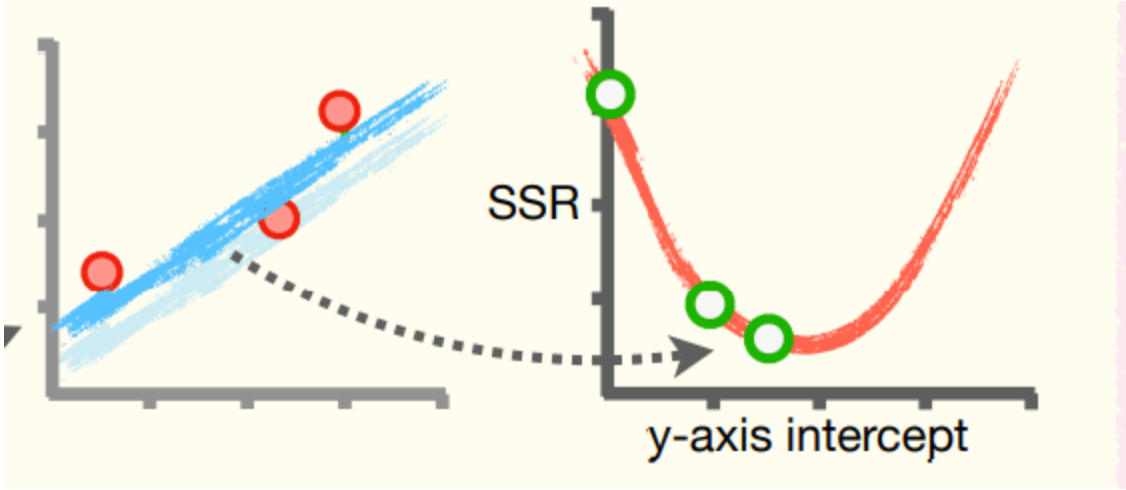
c. Calculate the new intercept value…

New intercept = Current intercept - Step Size

New intercept = 0.57 - (-0.23)

New intercept = 0.8

- The new intercept, 0.8, moves the line up a little closer to the data… …and it results in a lower SSR as shown below



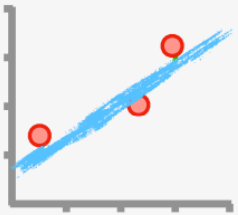
After **7** iterations…

a Evaluate the derivative at the current value…

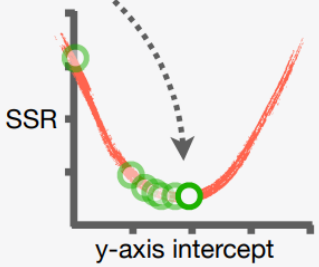
b Calculate the **Step Size**…

c Calculate the new value…

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



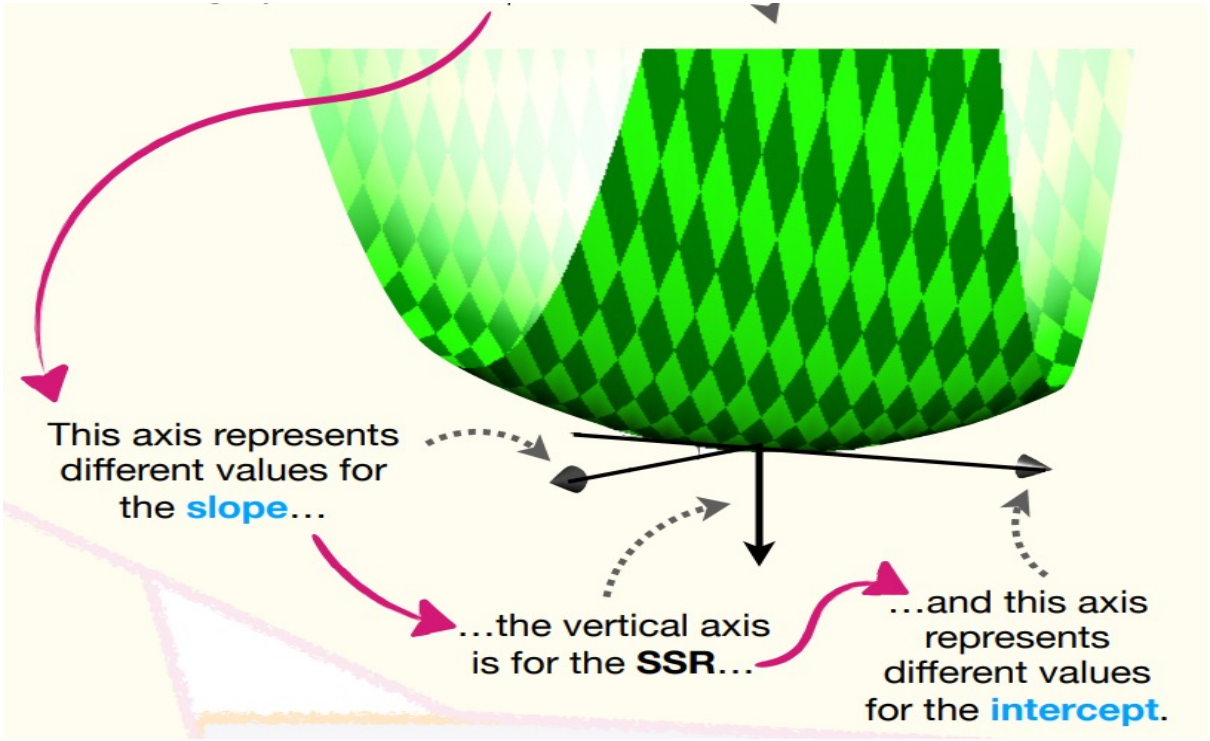
…and we made it to the lowest **SSR**.



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.

Optimizing Two or More Parameters: Detail

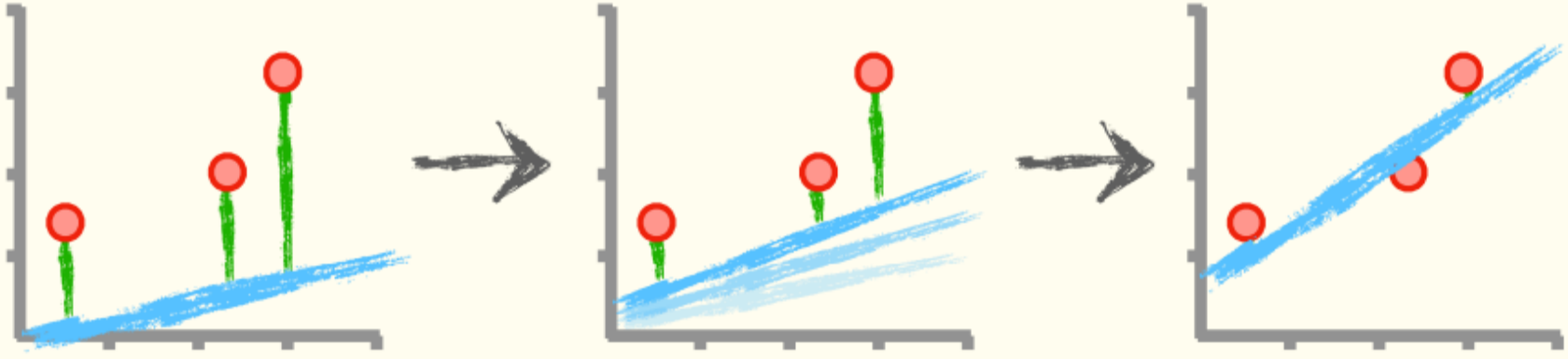
- Now that we know how to optimize the intercept of the line that minimizes the SSR, let's optimize both the intercept and the slope
- When we optimize two parameters, we get a 3-dimensional graph of the SSR



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

  SSR = (Height - (intercept + slope x Weight))^2

- 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*.



- As in the case of intercept we can use the Chain Rule to tell us how the SSR changes with respect to the slope.

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

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

Residual = Observed Height - (intercept + slope x Weight)

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

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

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

$$\frac{d\,Residual}{d\,slope} = \frac{d}{d\,slope}\,Height - (intercept + slope \times Weight)$$

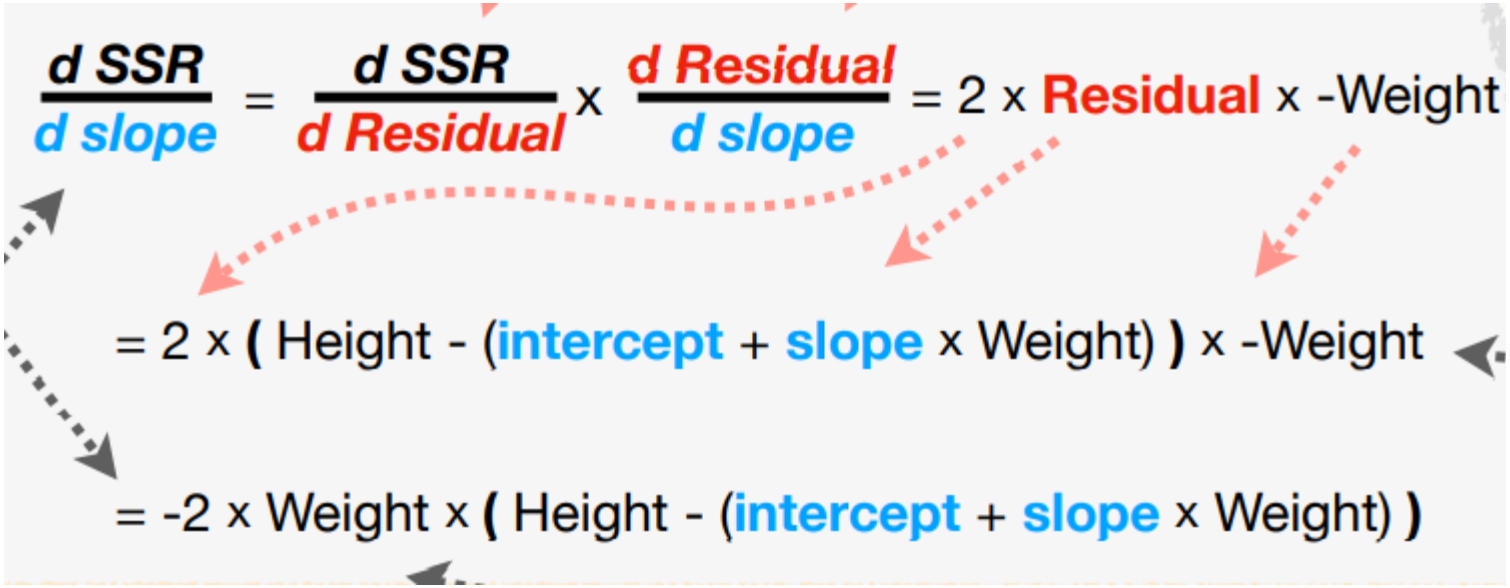
$$= \frac{d}{d\,slope}\,Height - intercept - slope \times Weight$$

$$= 0 - 0 - Weight = -Weight$$

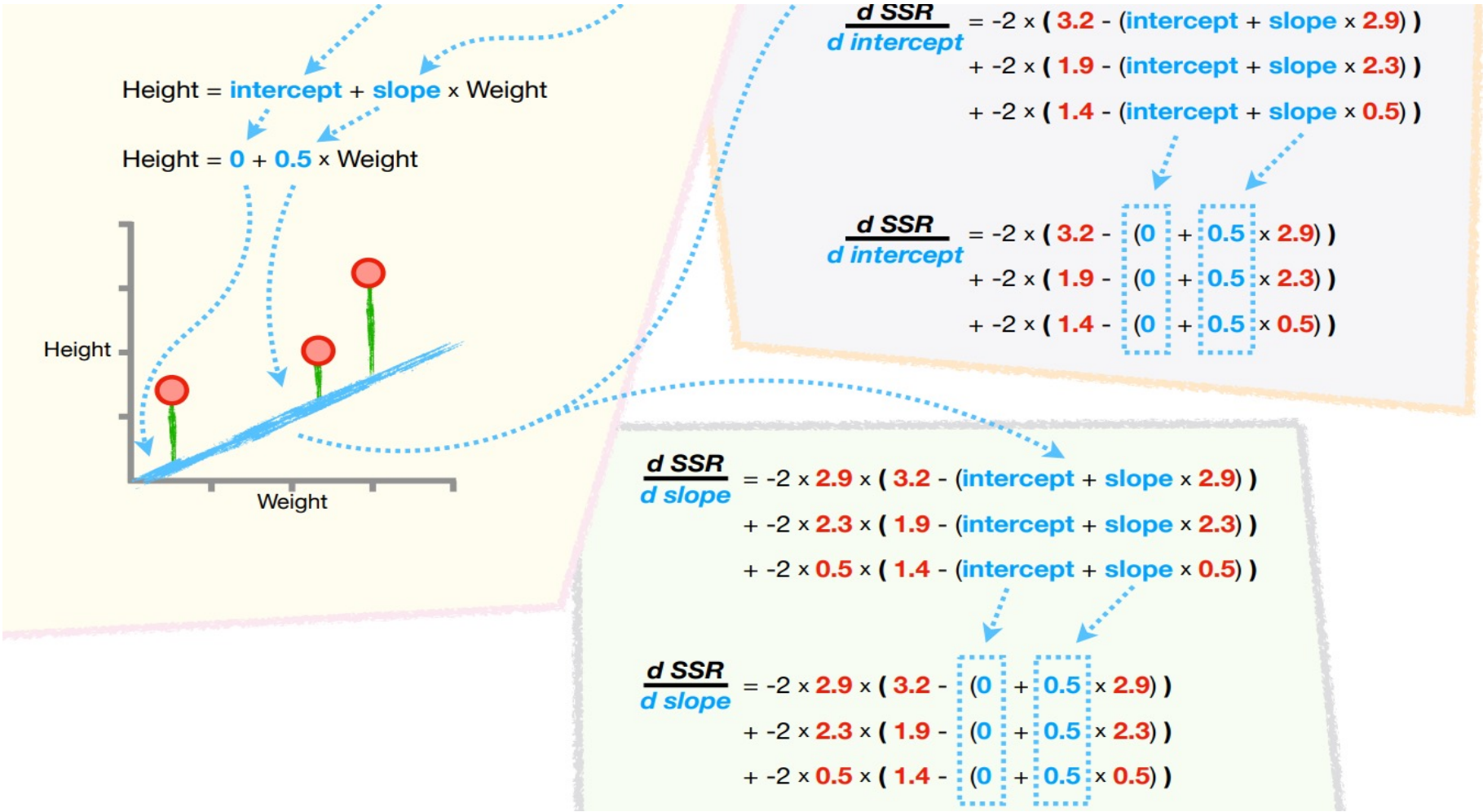
Because the first and second terms do not include the slope, their derivatives, with respect to the slope, are both **0**. However, the last term is the negative slope times **Weight**, so its derivative is **-Weight**.

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

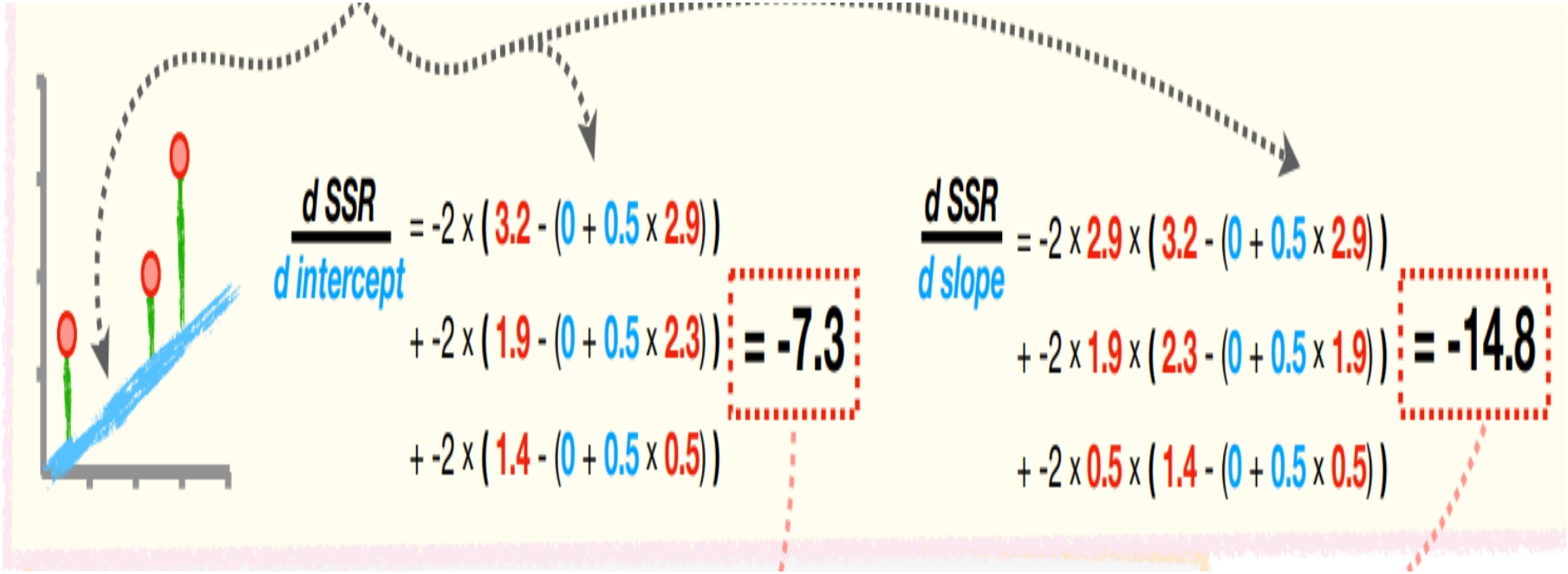
- Step 4: Plug the derivatives into The Chain Rule to get the final derivative of the SSR with respect to the slope.



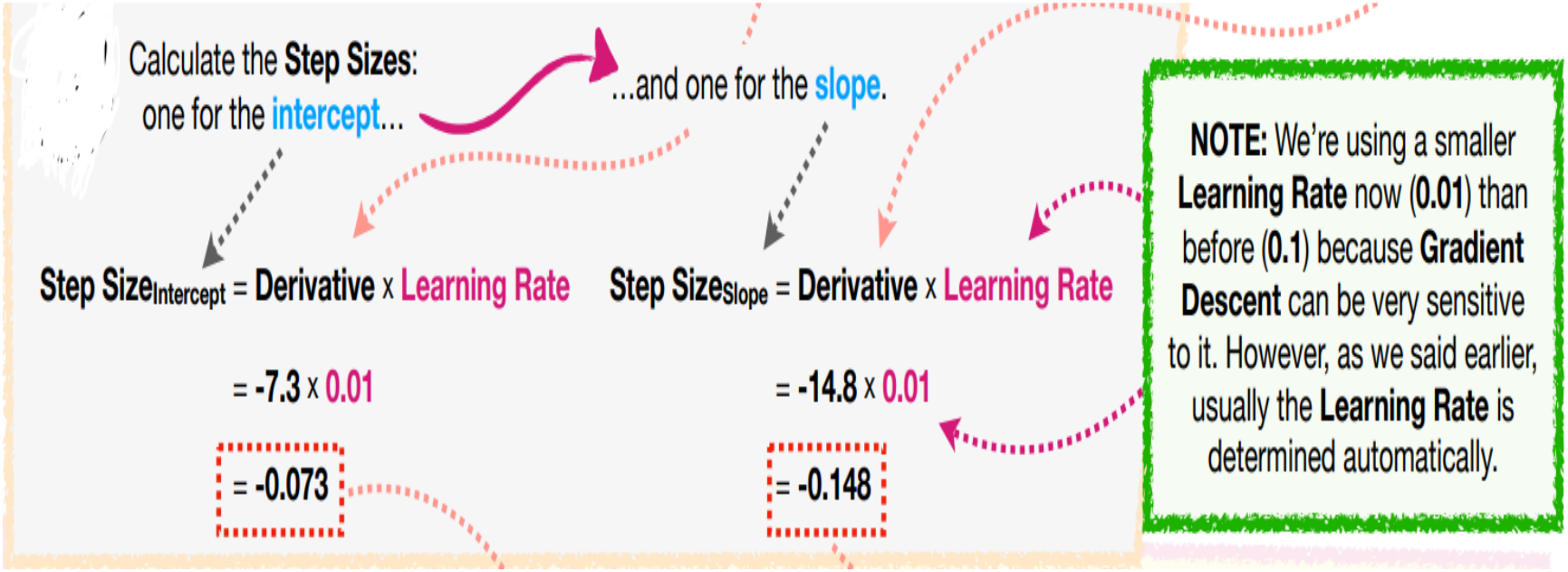
- Step 5: Now initialize the parameter, or parameters, that we want to optimize with random values. In this example, we'll set the intercept to 0 and the slope to 0.5



- Step 6: Evaluate the derivatives at the current values for the intercept, 0, and slope, 0.5



Step 7: Calculate the Step Sizes: one for the intercept……and one for the slope.



- Step 8: Take a step from the current intercept, 0, and slope, 0.5, to get closer to the optimal values…

  New intercept = Current intercept - Step SizeIntercept

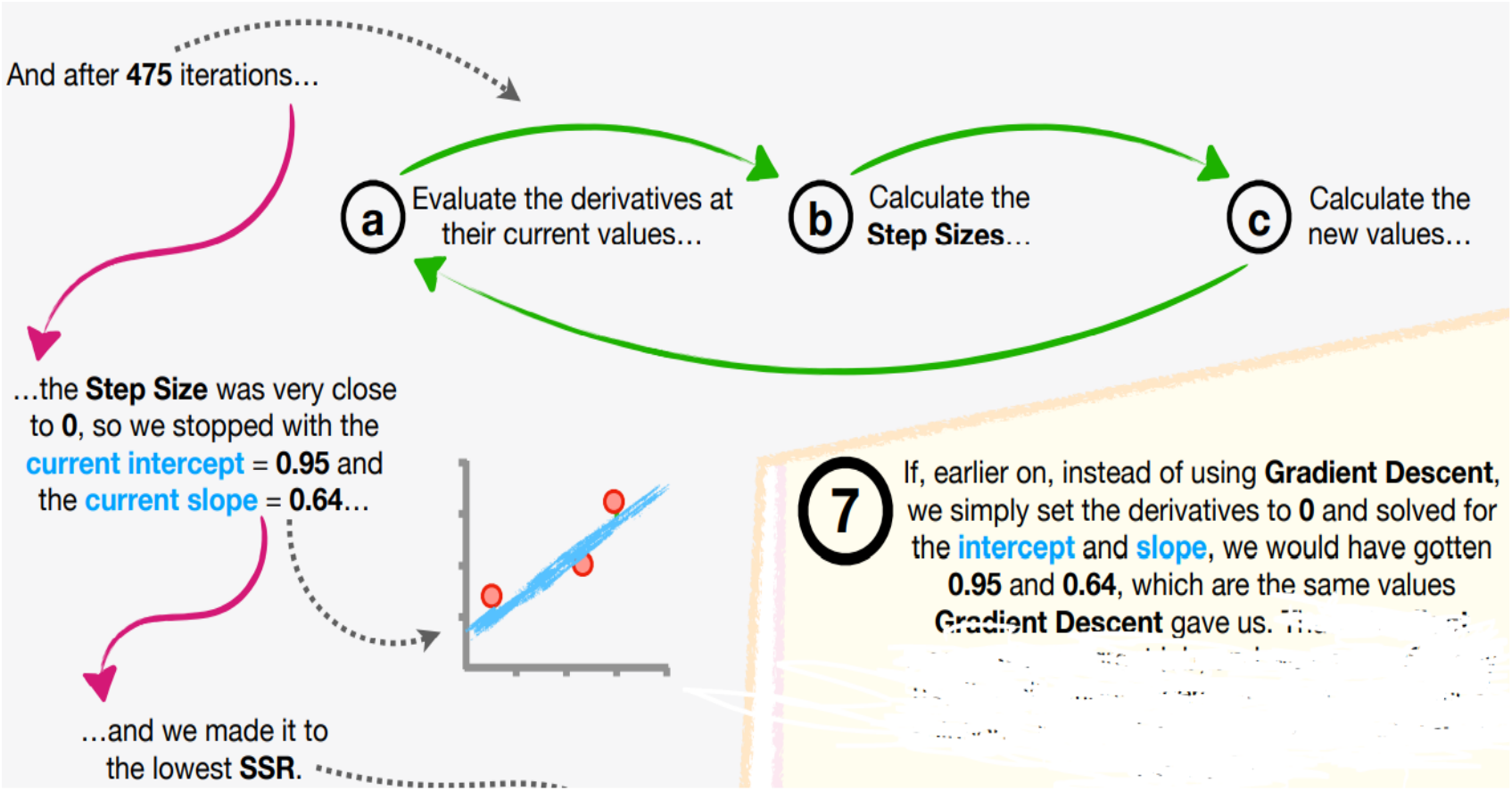
  New intercept = 0 - (-0.073) New intercept = 0.073

  …and the intercept increases from 0 to 0.073, the slope increases from 0.5 to 0.648, and the SSR decreases

  New slope = Current slope - Step SizeSlope

  = 0.5 - (-0.148)

  =New slope = 0.648


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.
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.