

Optimization & Stochastic Gradient Descent

Bibek Poudel

Sections

- Optimization
- Gradient Descent
- Problems with Gradient Descent
- Stochastic Gradient Descent
- Recap

Optimization

Optimization

- “Doing most with the least”
- “Find most effective or favorable values”
- E.g., minimize cost, maximize profit

Optimization

- Statistics, Machine Learning, Data Science → Solving optimization
- Mathematically:

“maximize or minimize a function by systematically choosing input values from an allowed set while fulfilling constraints, if any”

Optimization

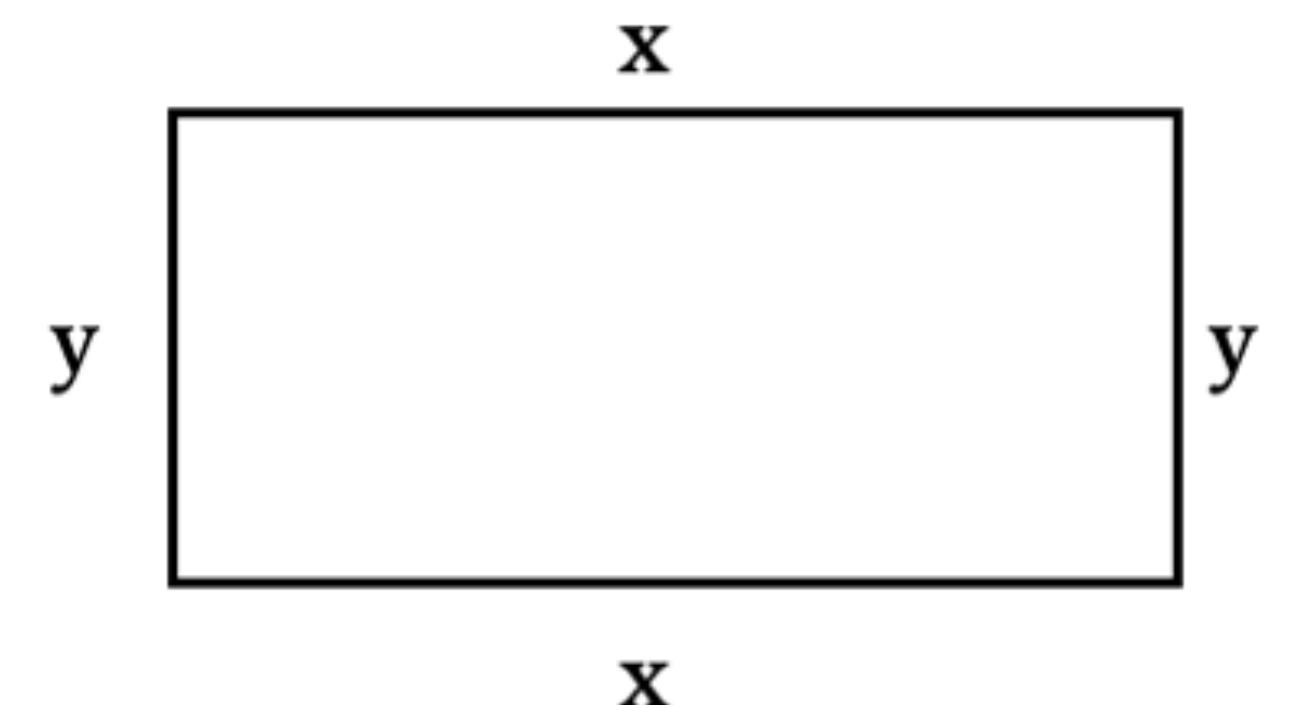
- Statistics, Machine Learning, Data Science → Solving optimization
- Mathematically:

“maximize or minimize a function by systematically choosing input values from an allowed set while fulfilling constraints, if any”

Optimization (Toy example)

Optimization (Toy example)

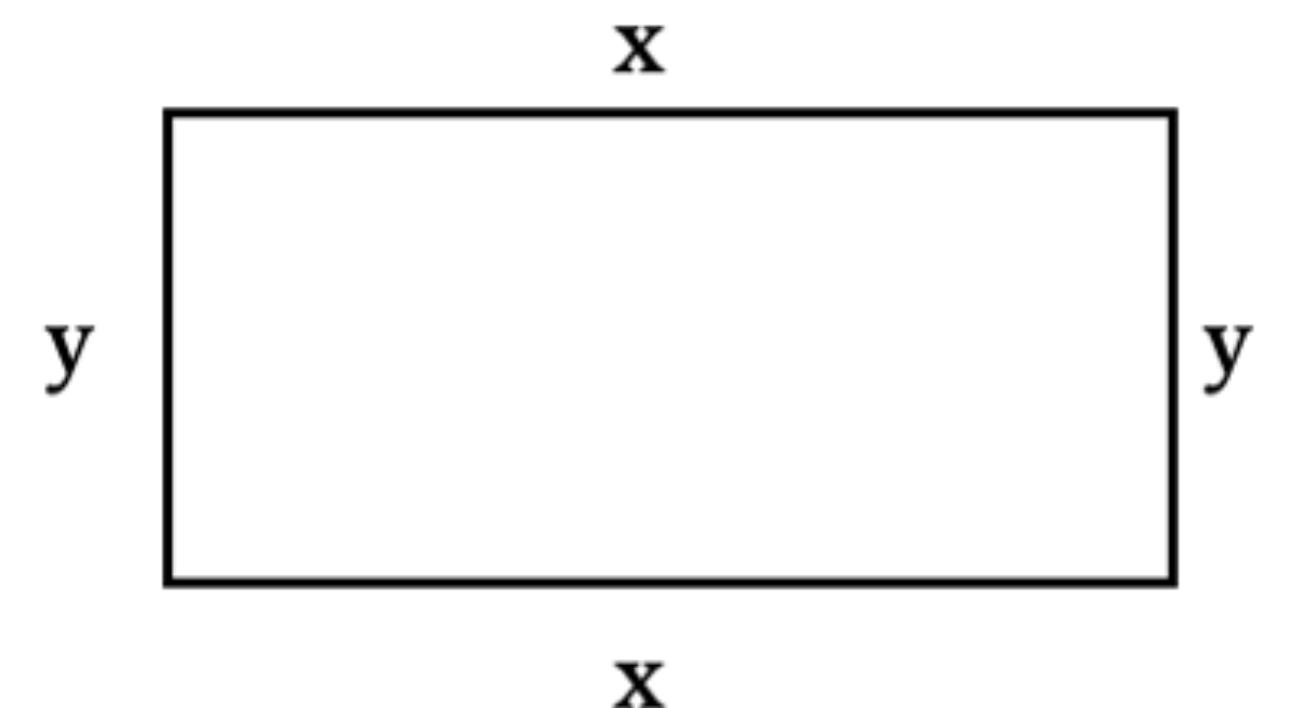
Find the length and breadth of a rectangle whose area is 100, keeping the perimeter as small as possible.



Optimization (Toy example)

- Objective Function
- Constraint

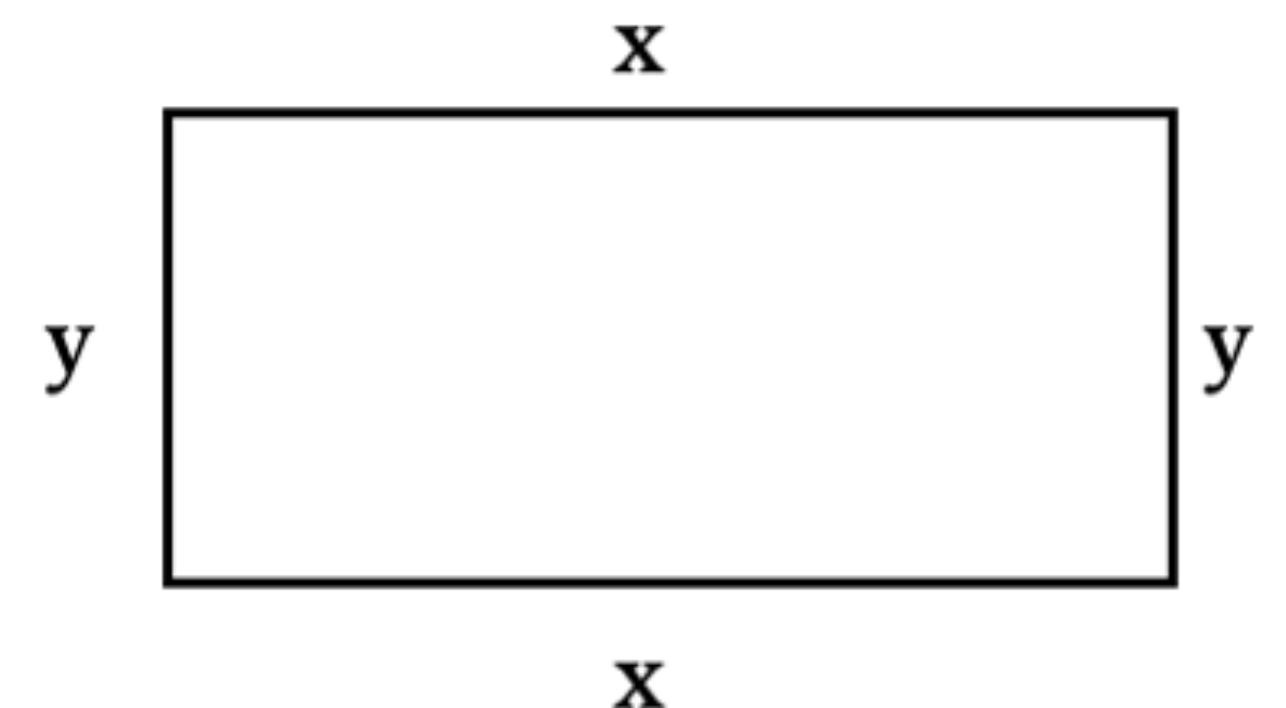
Find the length and breadth of a rectangle whose area is 100, keeping the perimeter as small as possible.



Optimization (Toy example)

- Objective Function
- Constraint

Find the length and breadth of a rectangle whose area is 100, keeping the perimeter as small as possible.



Minimize: $2(x+y)$
Subject to: $x \cdot y = 100$

Goal: Find the best fit of x and y
Solution: $x=10, y=10$

Optimization

- Hand crafting a solution may not always be feasible
- A dimensional problem may not be intuitive

Gradient Descent

Gradient Descent

- First order iterative algorithm to find a local minimum of a differentiable function.

Gradient Descent

- First order iterative algorithm to find a local minimum of a differentiable function.
- Key words:
 - ▶ First order
 - ▶ Iterative
 - ▶ Local minimum
 - ▶ Differentiable

Gradient Descent (Worked out example)

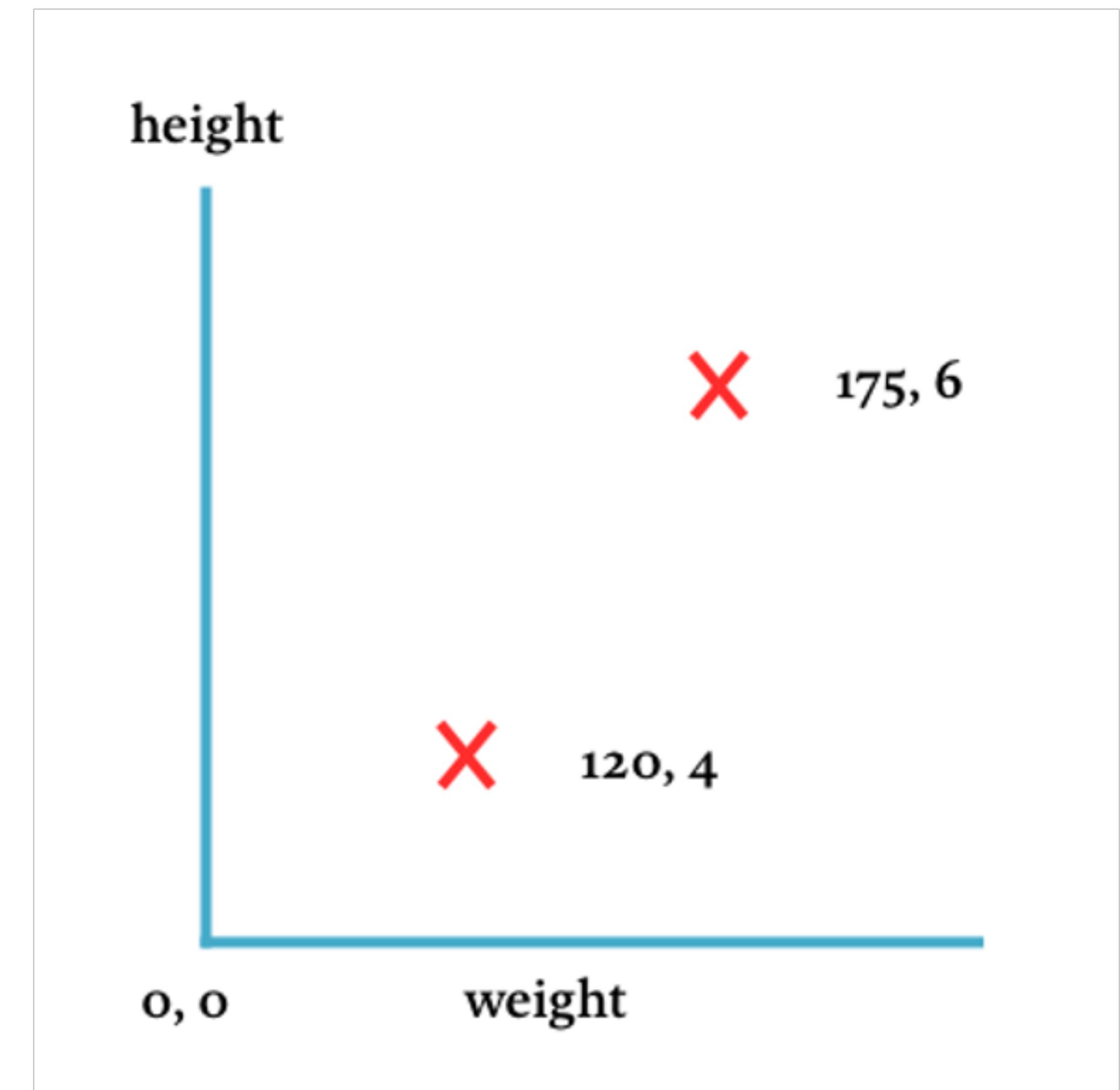
Gradient Descent (Worked out example)

- Data (feature, target)
- Goal

weight (lb.)	height (ft.)
120	4
175	6

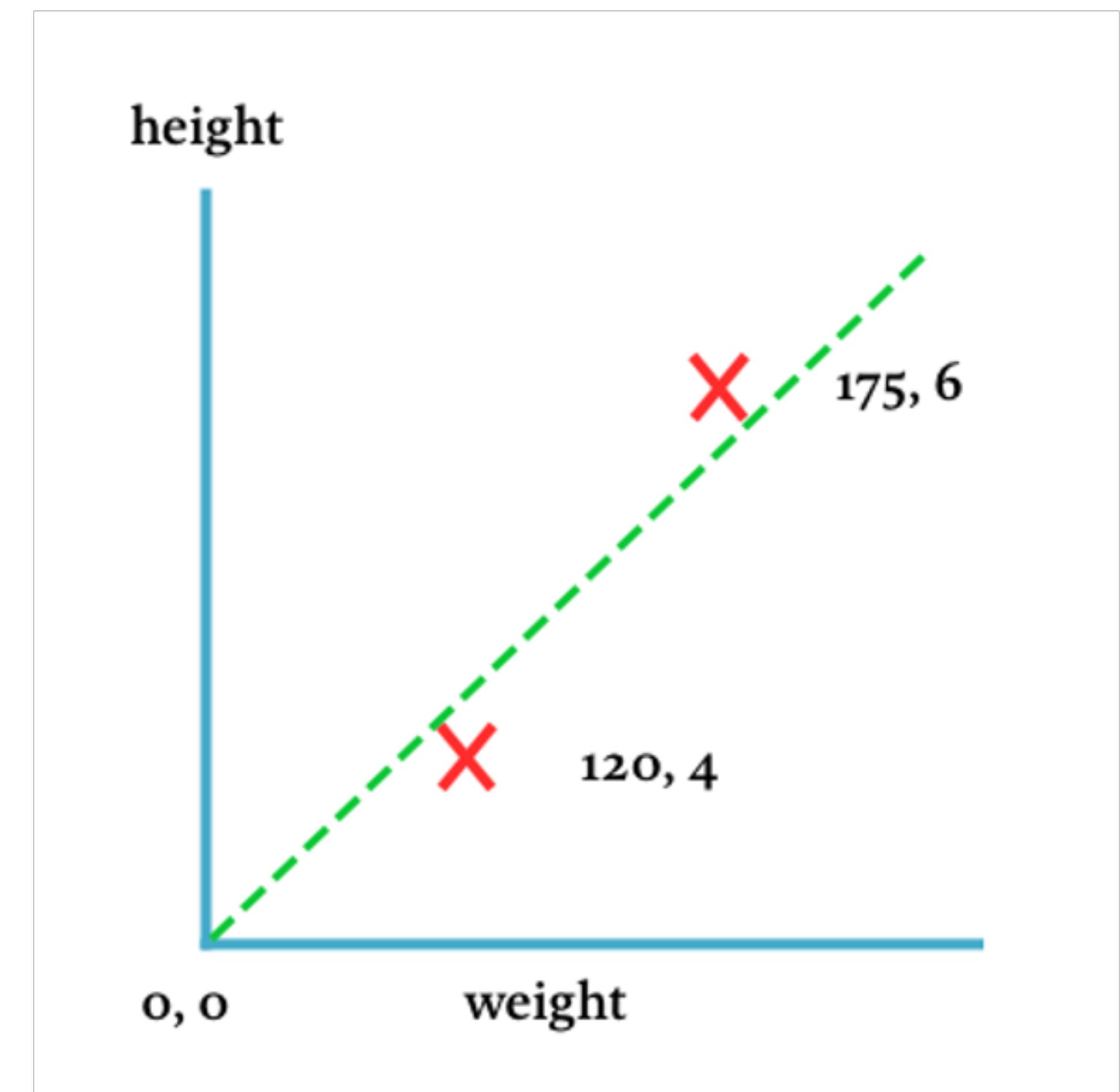
Gradient Descent (Worked out example)

- Data plot



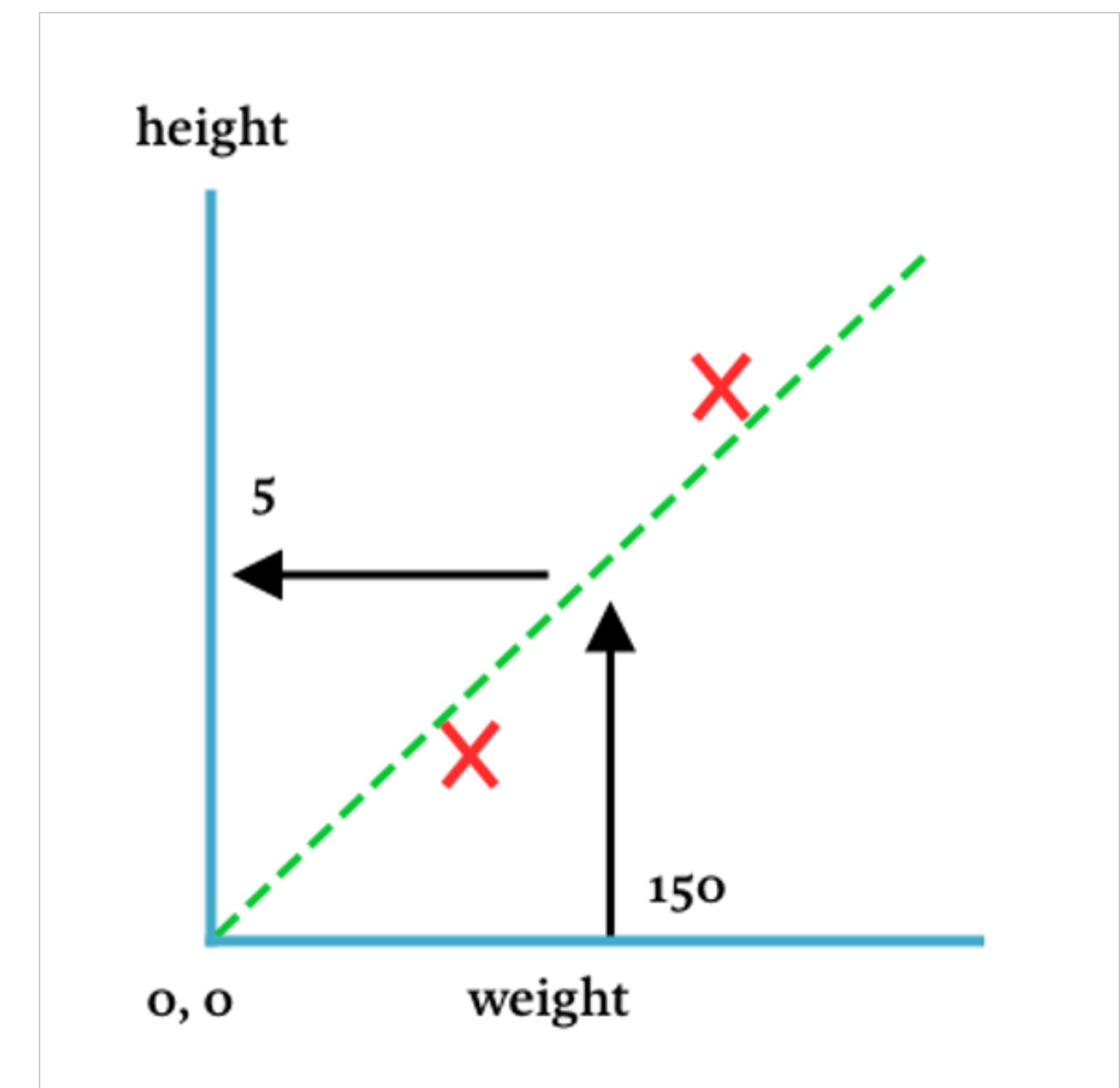
Gradient Descent (Worked out example)

- Fitted model



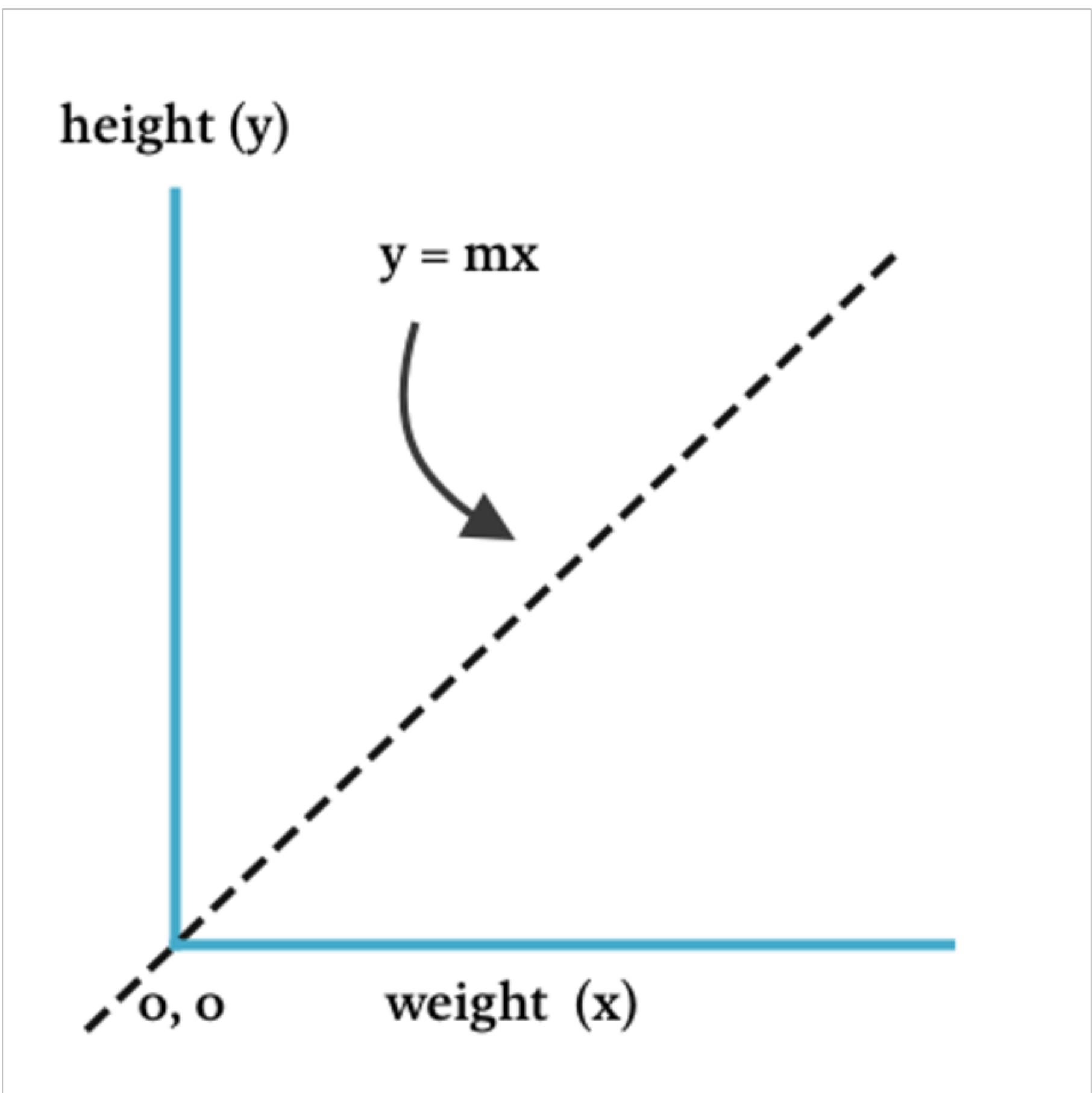
Gradient Descent (Worked out example)

- Make inference



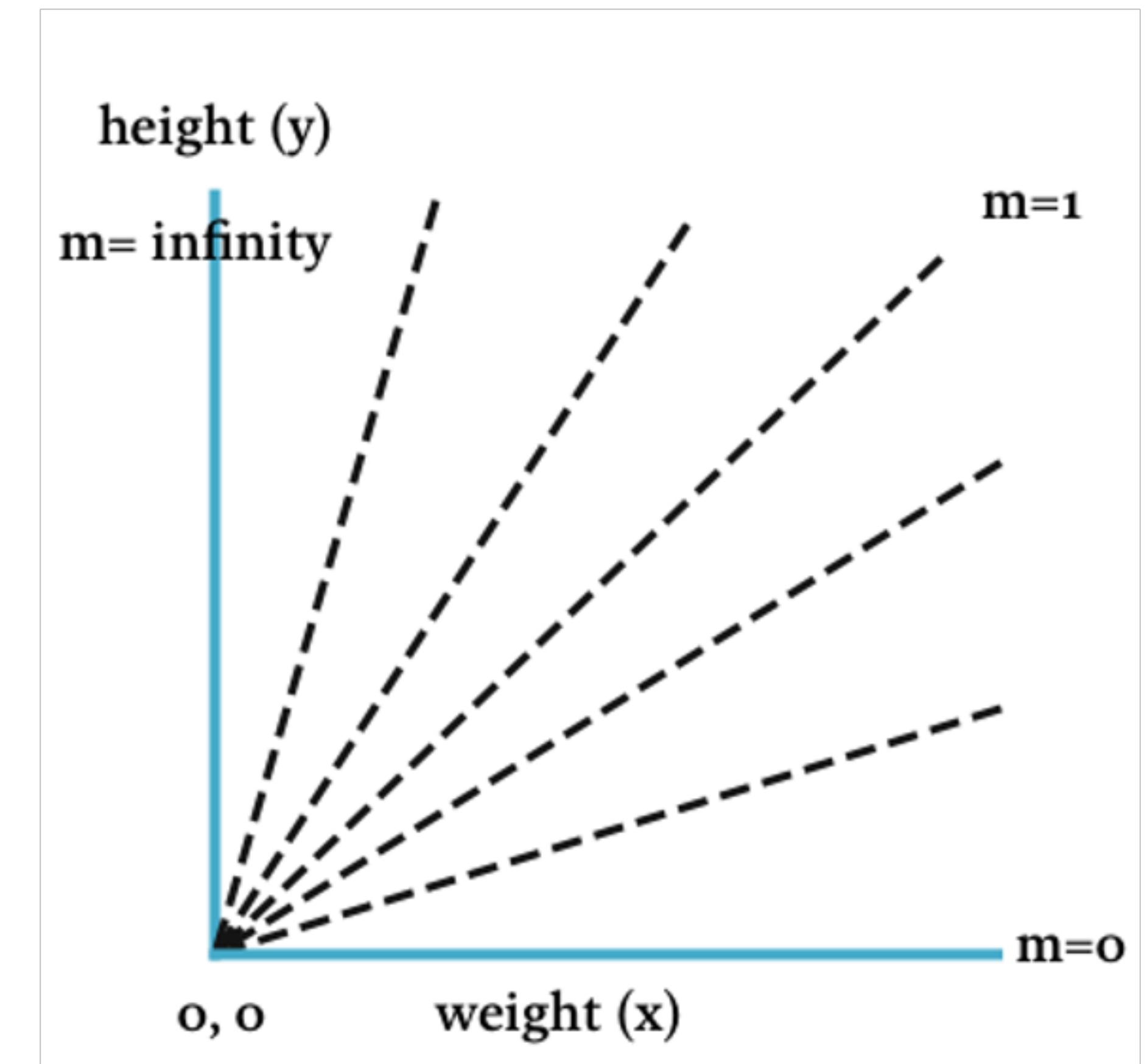
Gradient Descent (Worked out example)

- Linear model basics
 - ▶ Simplest equation of a line



Gradient Descent (Worked out example)

- Linear model basics
 - ▶ Parameter = slope (m)
 - ▶ Fit/ learn a model = find best ‘ m ’

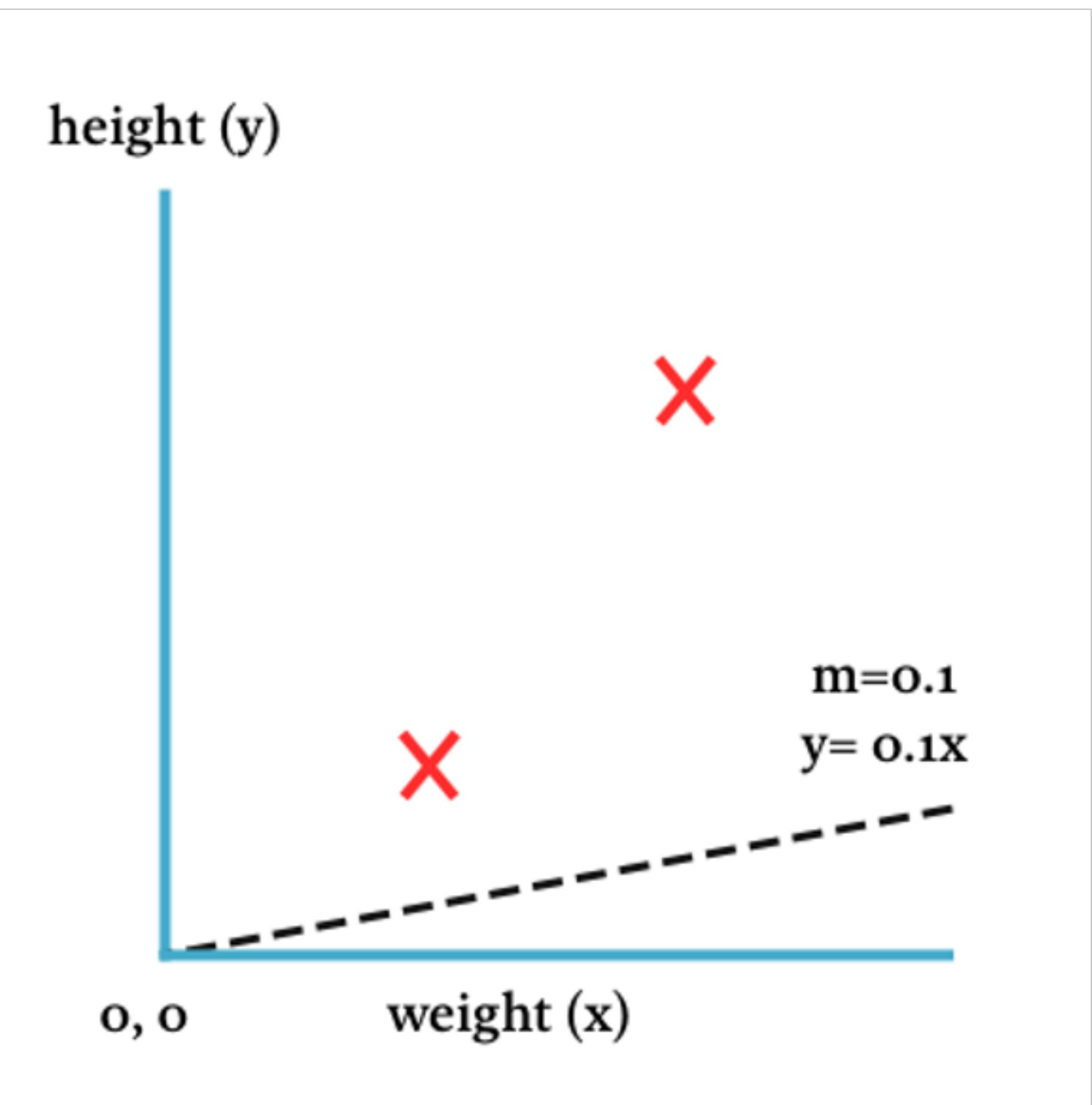


Gradient Descent (Worked out example)

- Start

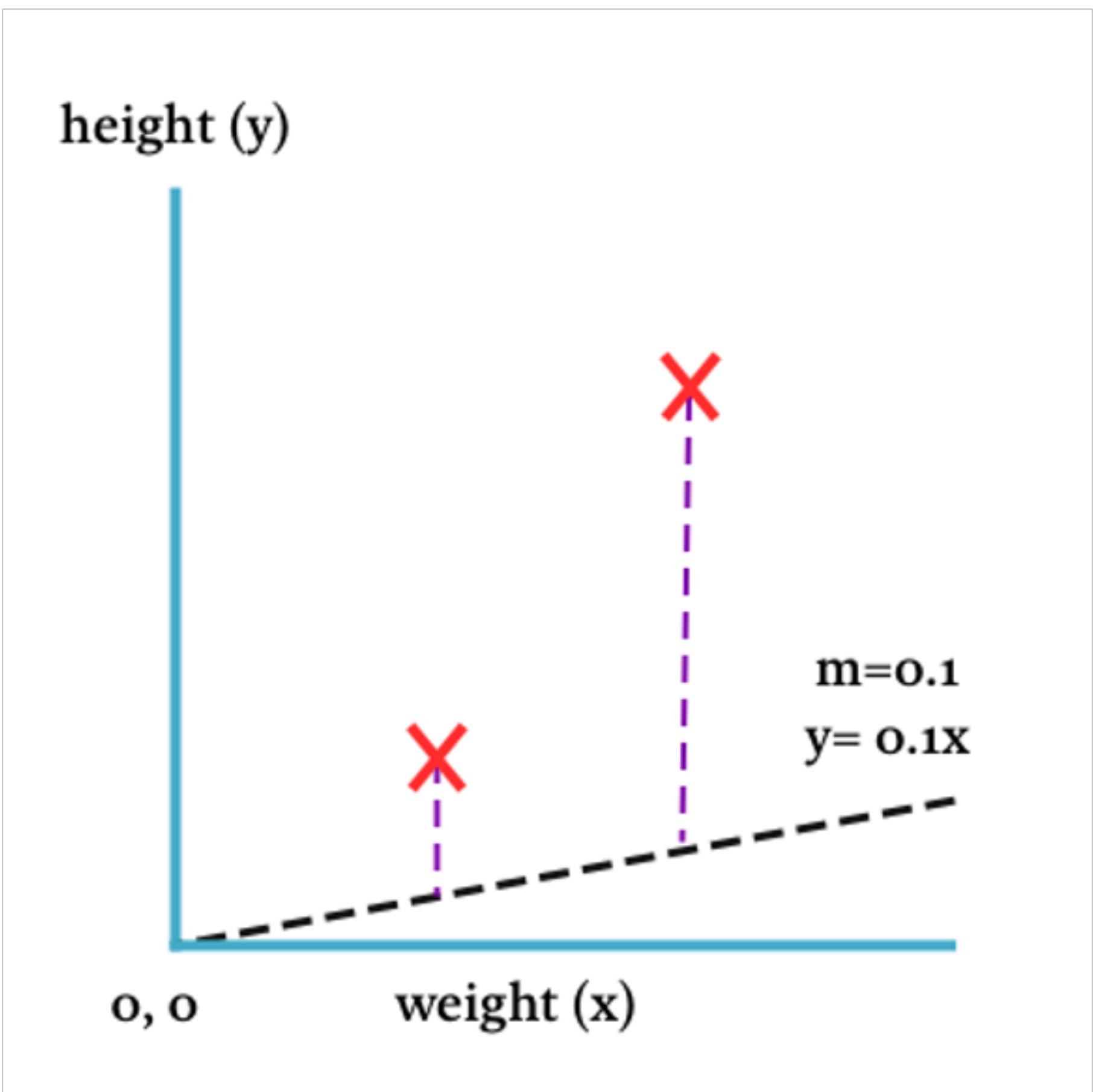
Gradient Descent (Worked out example)

- Step 1: Random initialization
 - ▶ $m = 0.1$



Gradient Descent (Worked out example)

- Step 1: Random initialization
 - ▶ $m = 0.1$
 - ▶ Does not fit well



Gradient Descent (Worked out example)

- Need to improve current parameter value

Gradient Descent (Worked out example)

- Need to improve current parameter value
- But... Before we improve it

Gradient Descent (Worked out example)

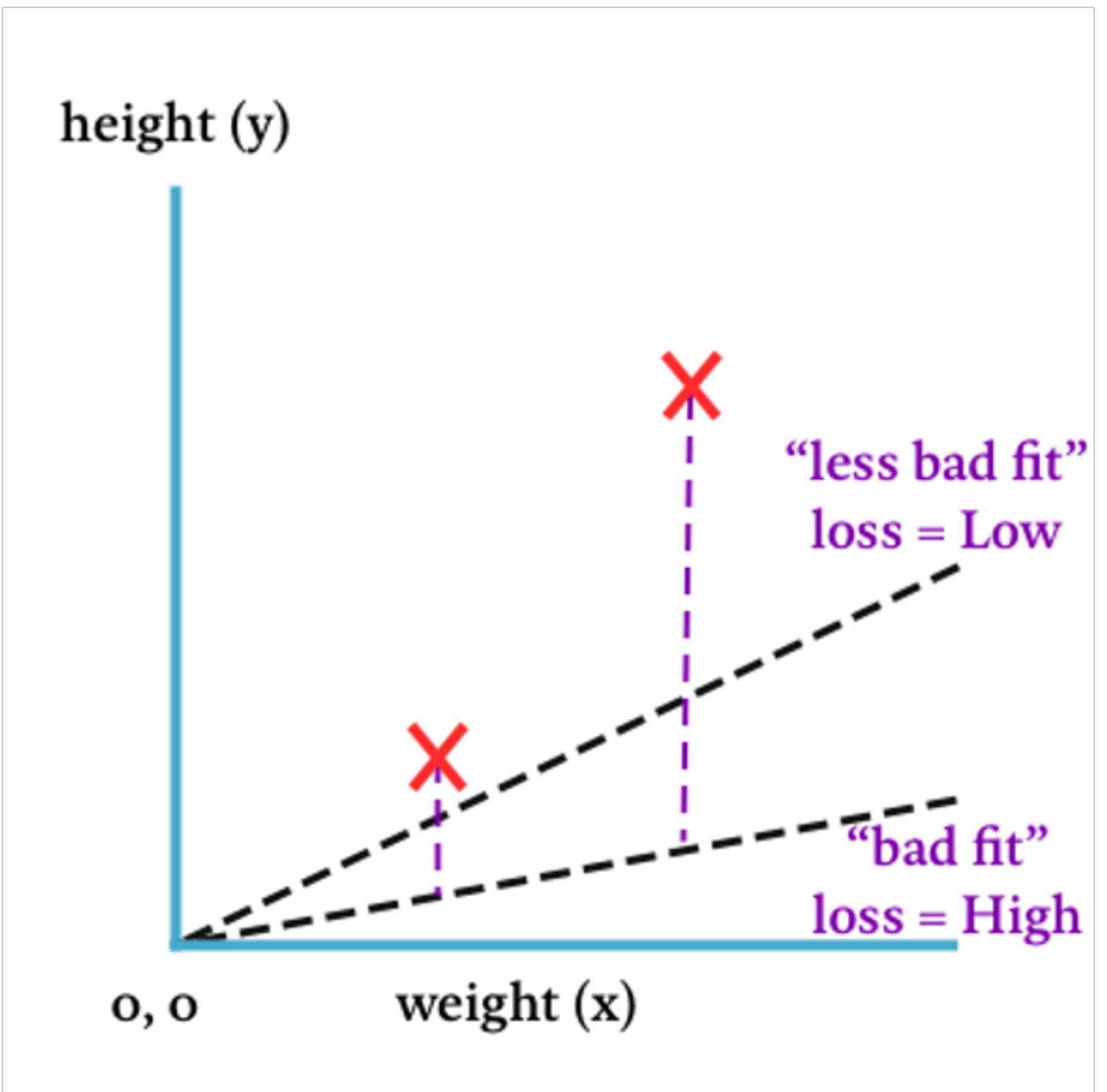
- Need to improve current parameter value
- But... Before we improve it
- We need to measure how “**good**” it is

Gradient Descent (Worked out example)

- Step 2: Define a Loss/ Error/ Cost function
 - ▶ $\text{Loss} = f(\text{parameters})$

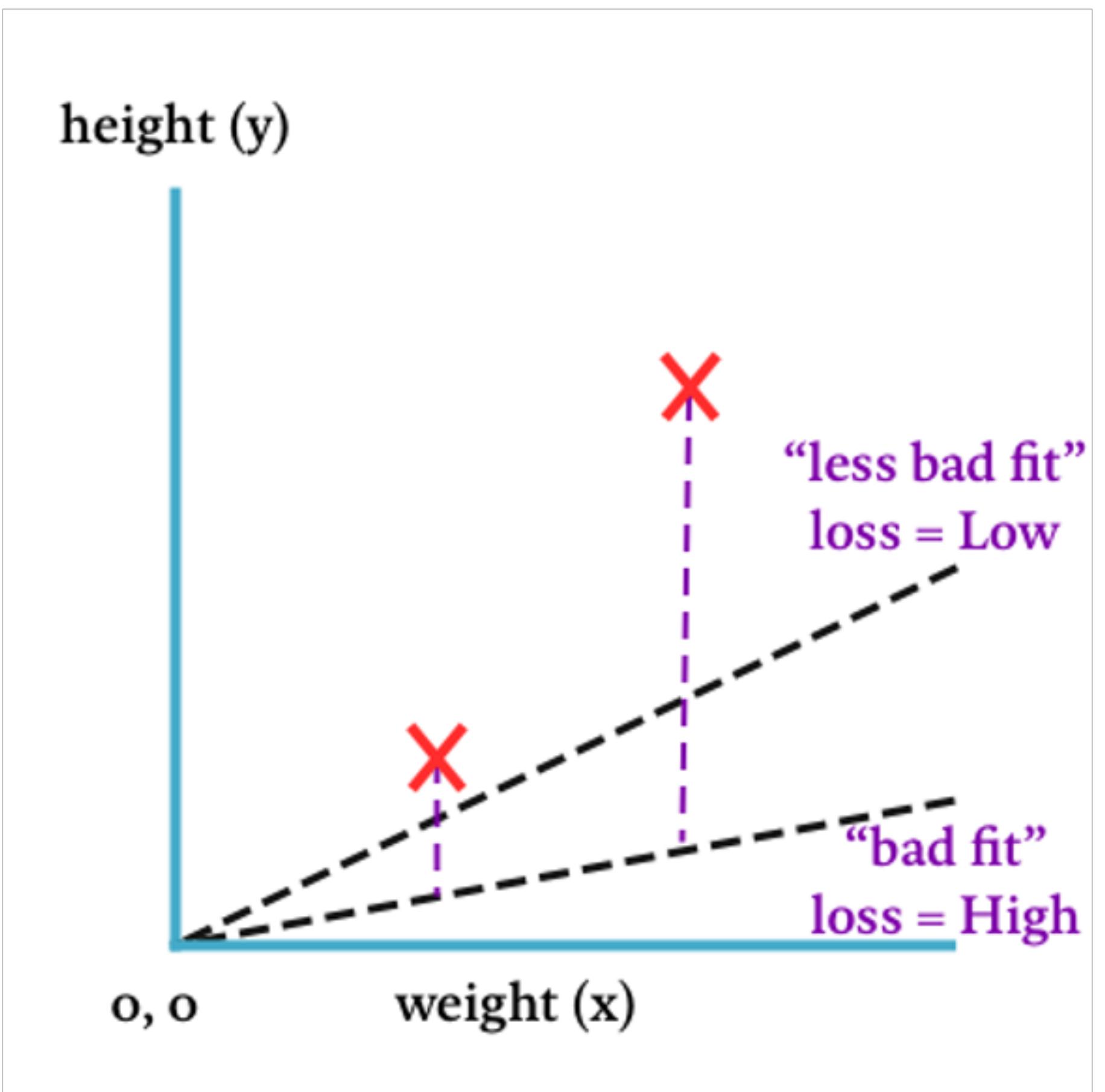
Gradient Descent (Worked out example)

- Step 2: Define a Loss/ Error/ Cost function
 - ▶ Loss = f (parameters)



Gradient Descent (Worked out example)

- Step 2: Define a Loss/ Error/ Cost function
 - ▶ Loss = f (parameters)
 - ▶ Does any loss function work?

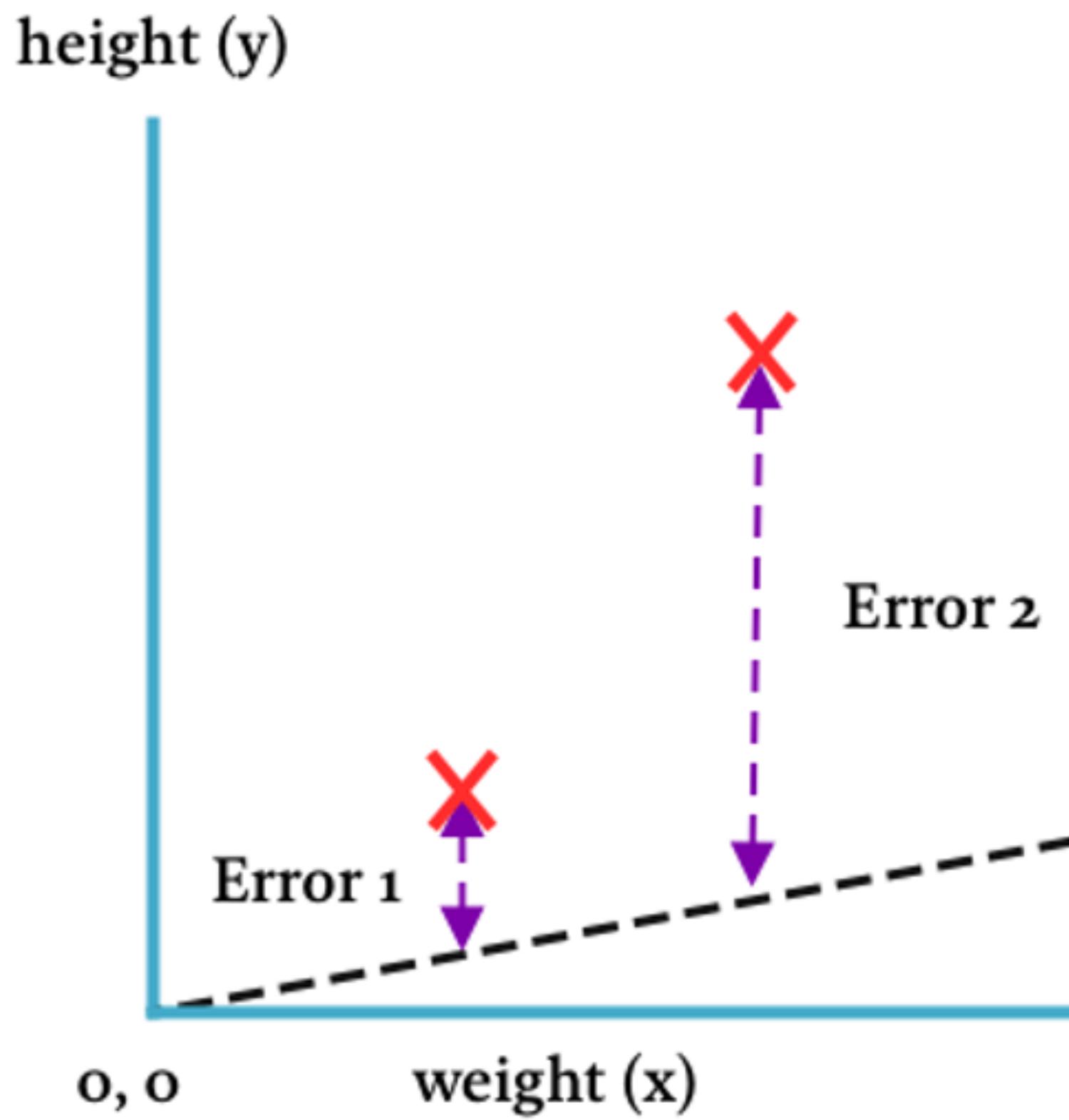


Gradient Descent (Worked out example)

- Step 2: Define a Loss/ Error/ Cost function

- ▶ Regression → Mean Squared Error

$$\text{MSE} = \frac{(\text{Error 1})^2 + (\text{Error 2})^2}{2}$$



Gradient Descent (Worked out example)

- Step 2: Define a Loss/ Error/ Cost function

► MSE calculation

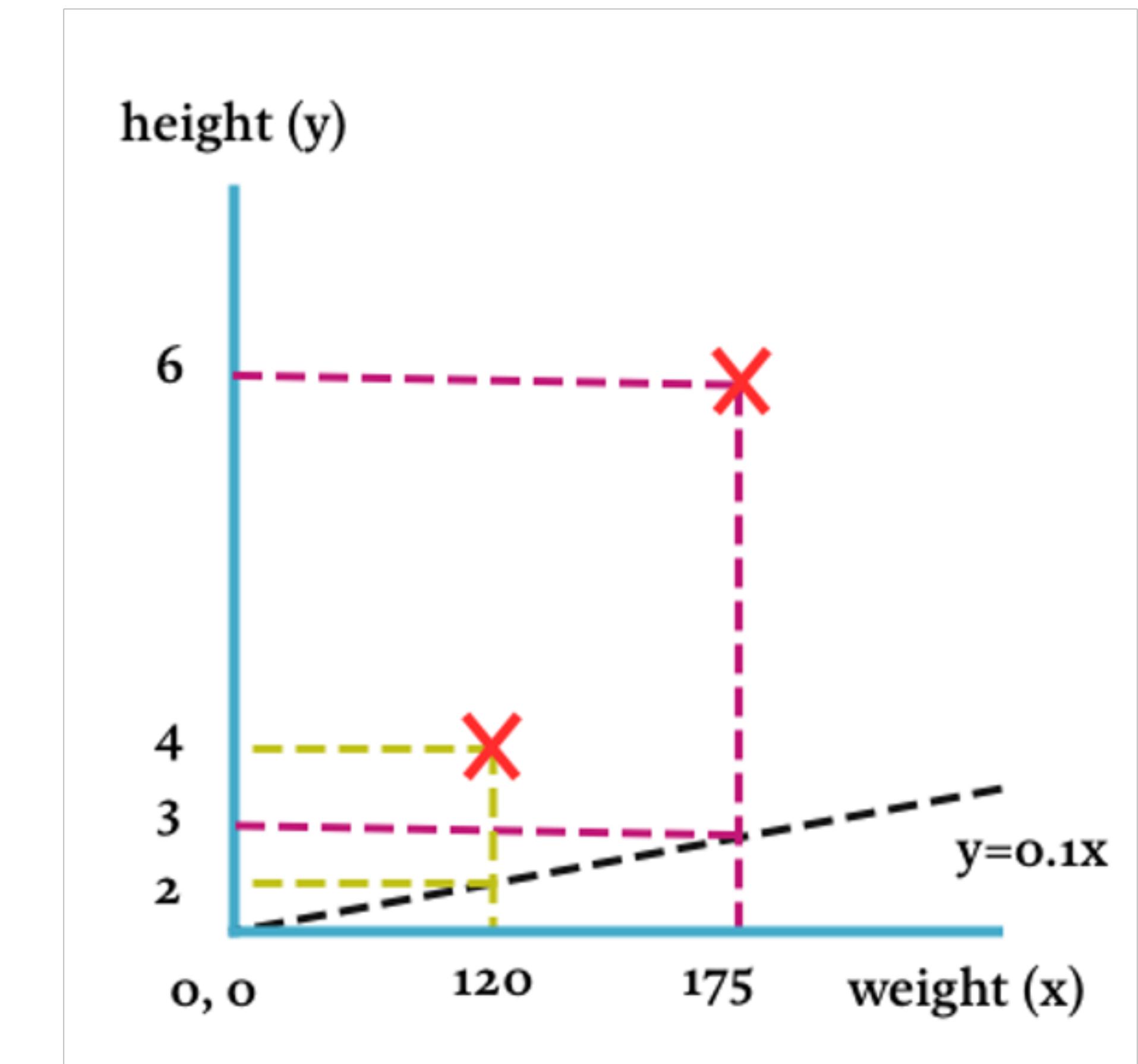
$$\text{Error 1} = (4 - 2) = 2$$

$$\text{Error 2} = (6 - 3) = 3$$

$$\text{MSE} = \frac{(\text{Error 1})^2 + (\text{Error 2})^2}{2}$$

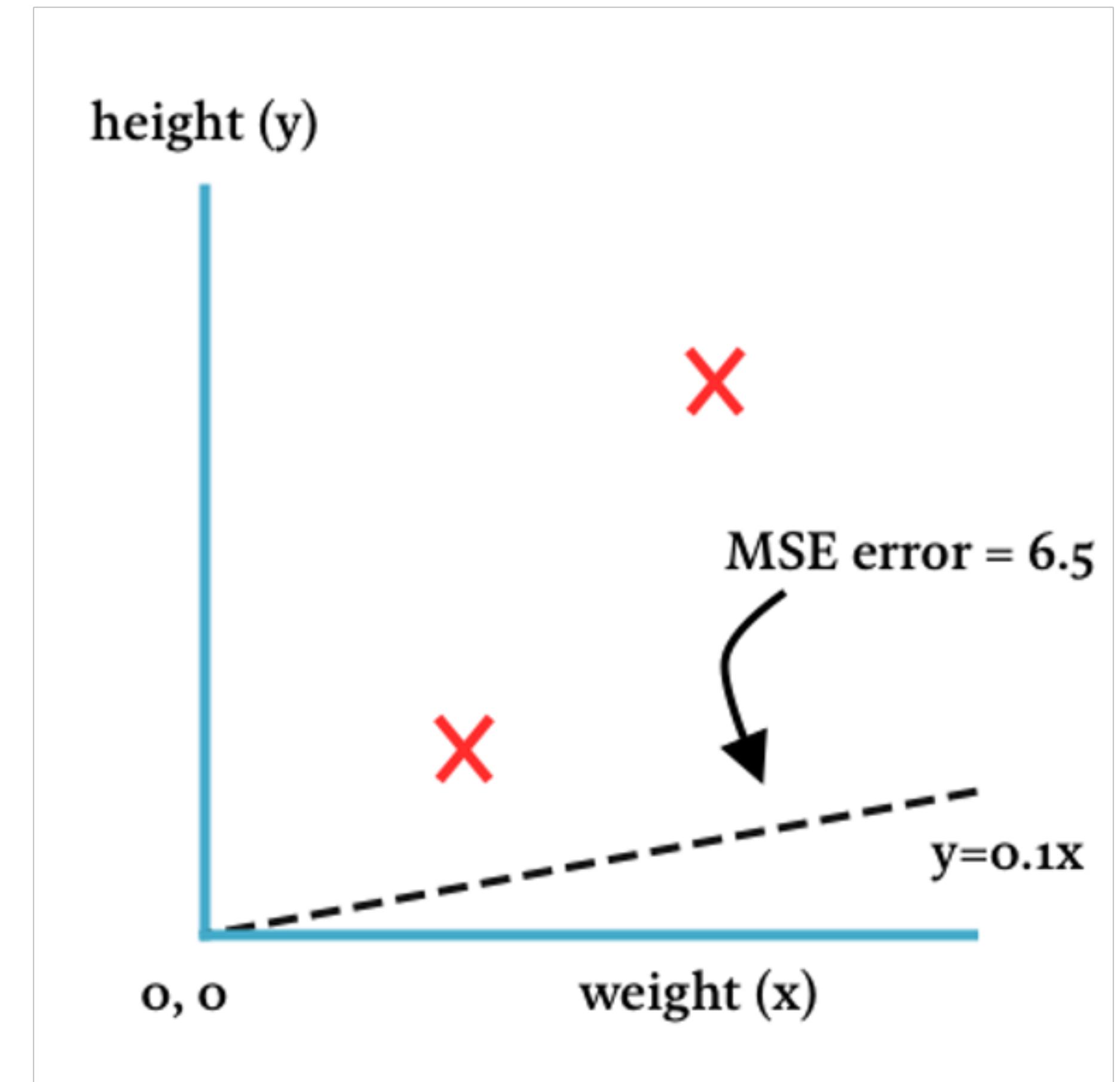
$$\text{MSE} = \frac{(2)^2 + (3)^2}{2}$$

$$\text{MSE} = 6.5$$



Gradient Descent (Worked out example)

- Step 2: Define a Loss/ Error/ Cost function
 - ▶ MSE calculation

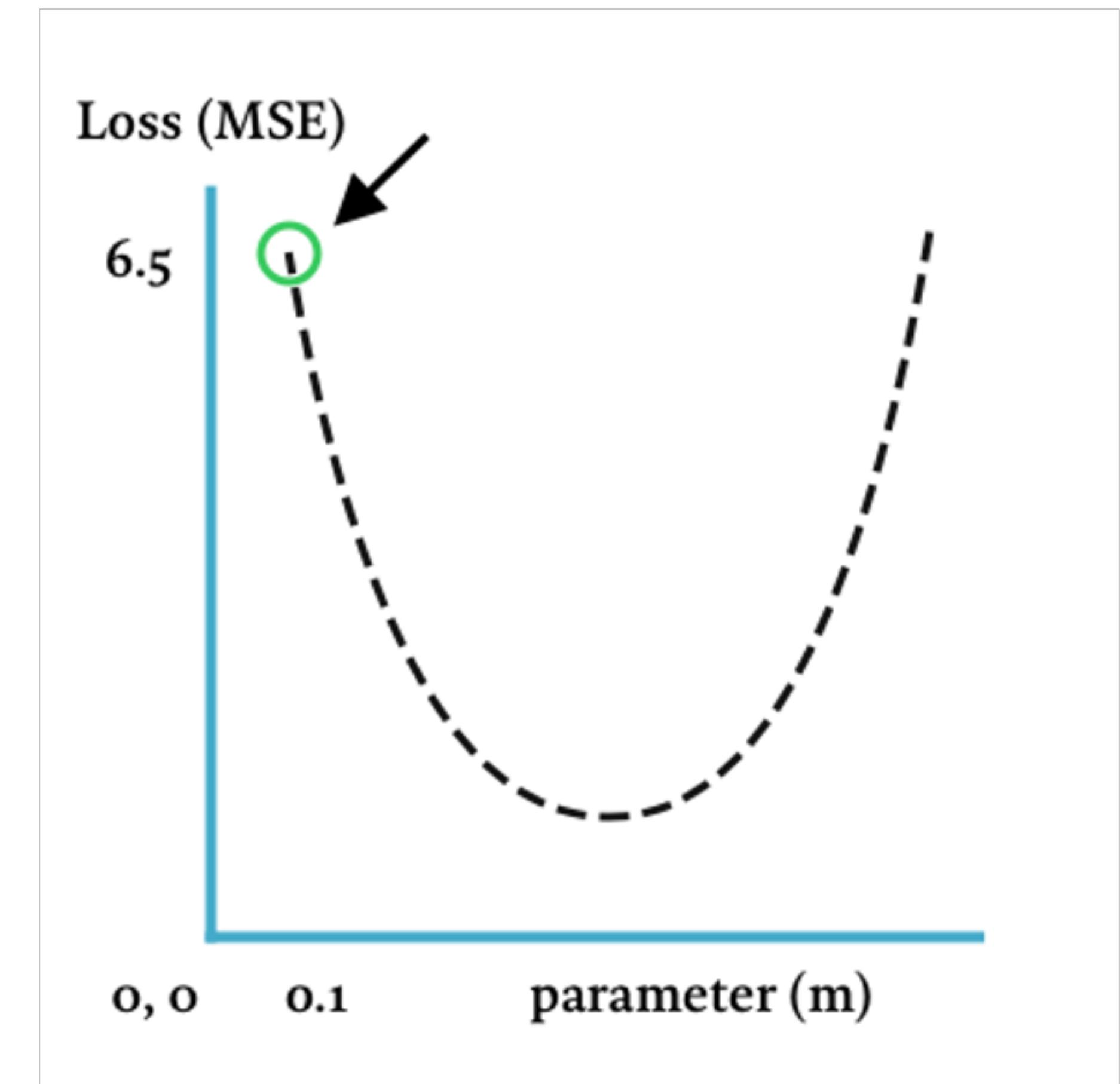


Gradient Descent (Worked out example)

- Now we improve
 - ▶ Gradient = First derivative
 - ▶ Descent = Move in decreasing direction

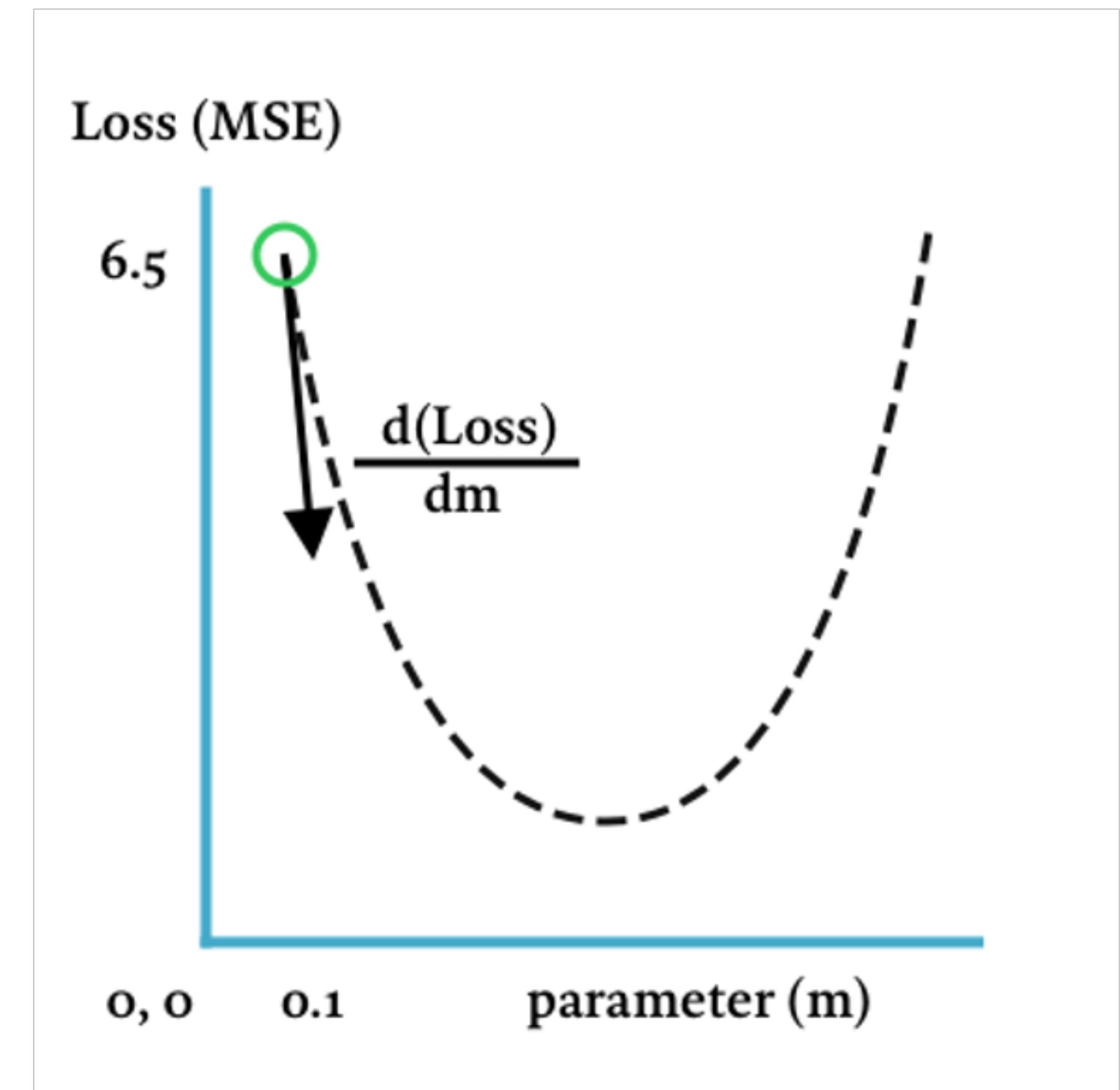
Gradient Descent (Worked out example)

- Now we improve
 - ▶ Gradient = First derivative
 - ▶ Descent = Move in decreasing direction



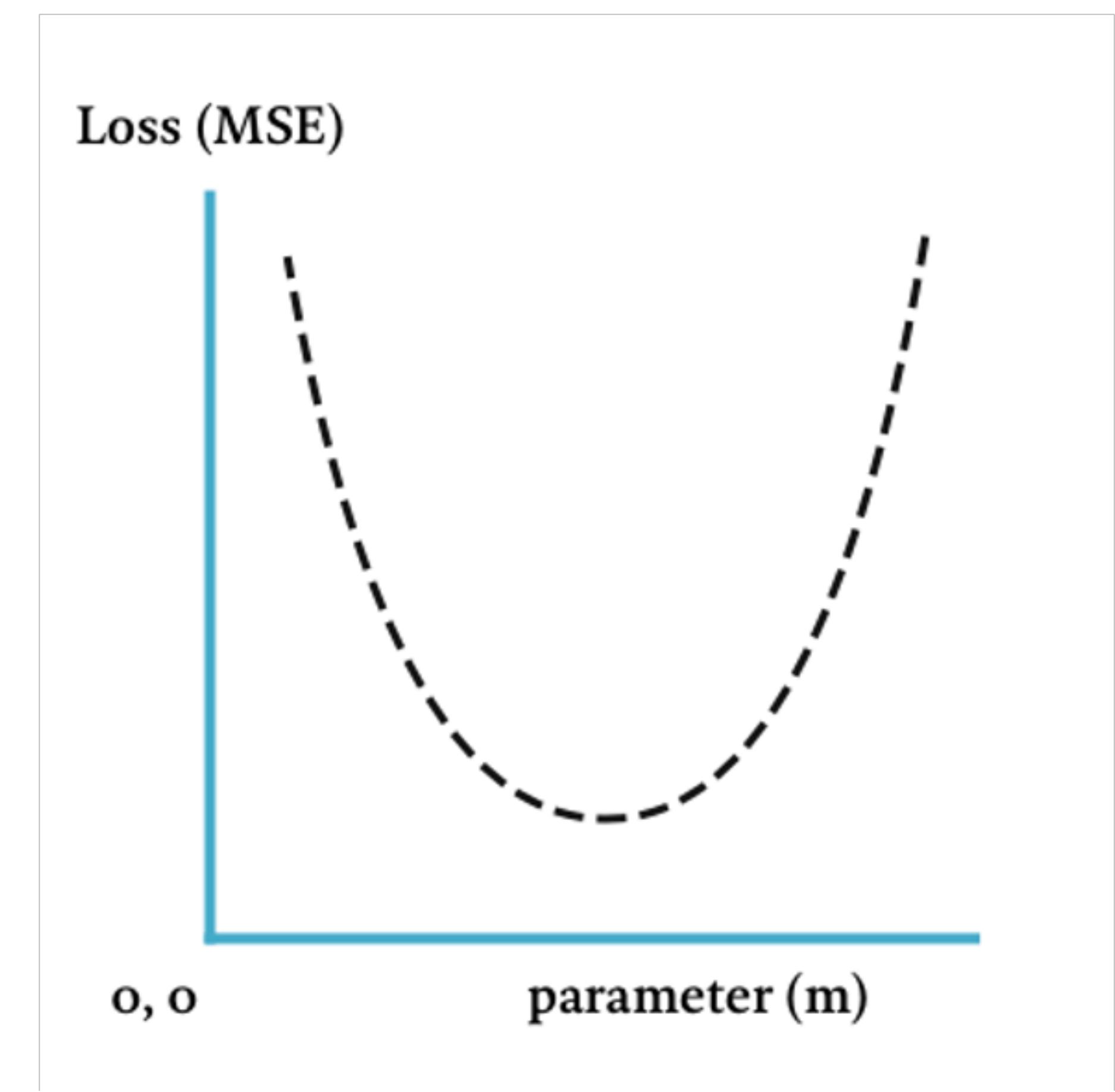
Gradient Descent (Worked out example)

- Now we improve
 - ▶ Gradient = First derivative
 - ▶ Descent = Move in decreasing direction



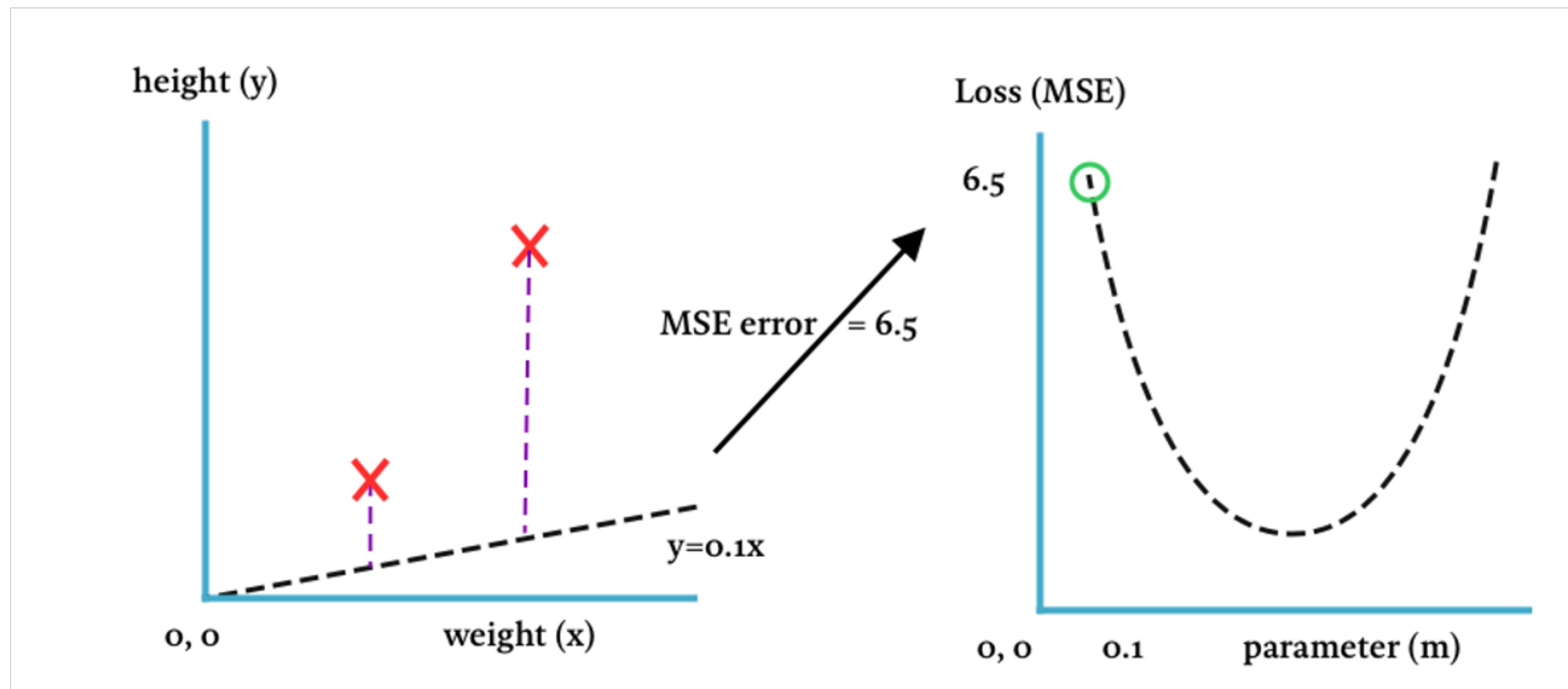
Gradient Descent (Worked out example)

- Why is the curve convex shaped?



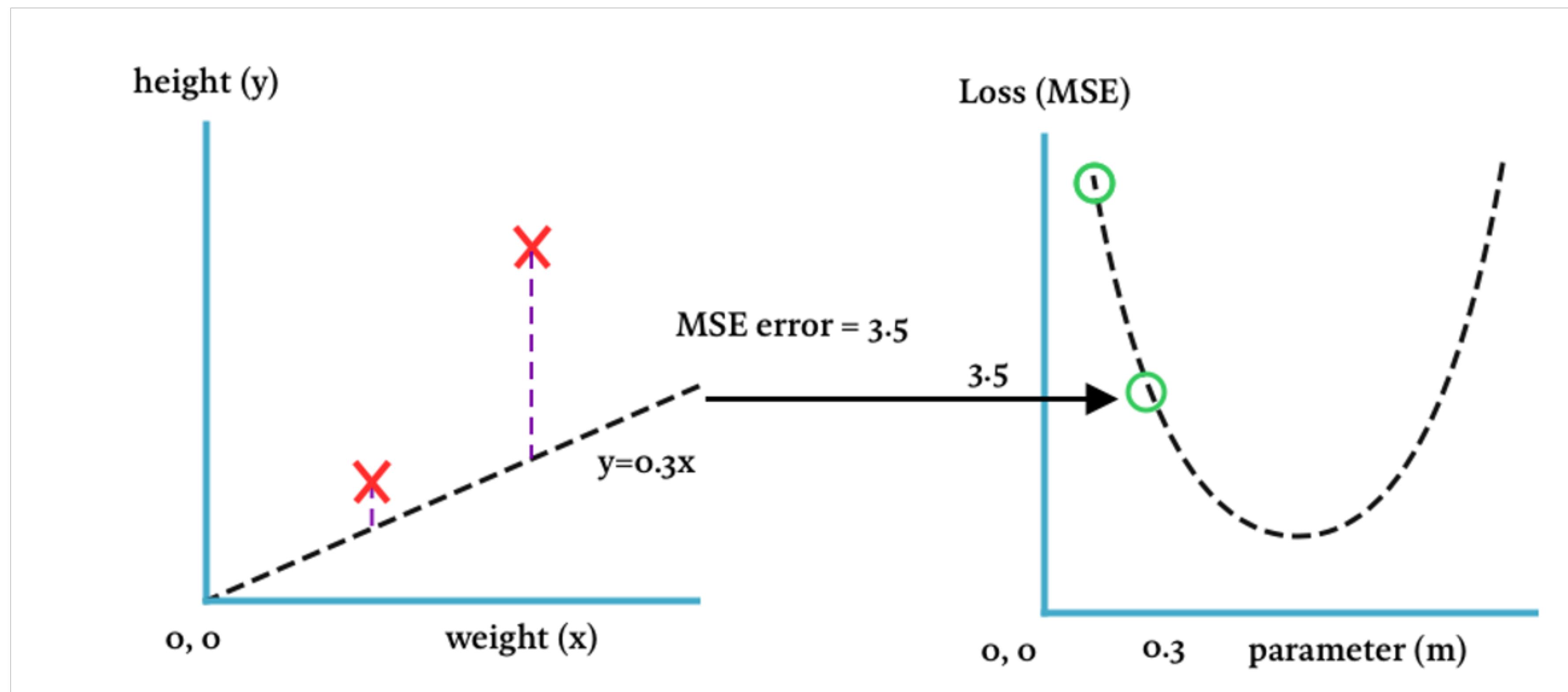
Gradient Descent (Worked out example)

- Why is the curve convex shaped?



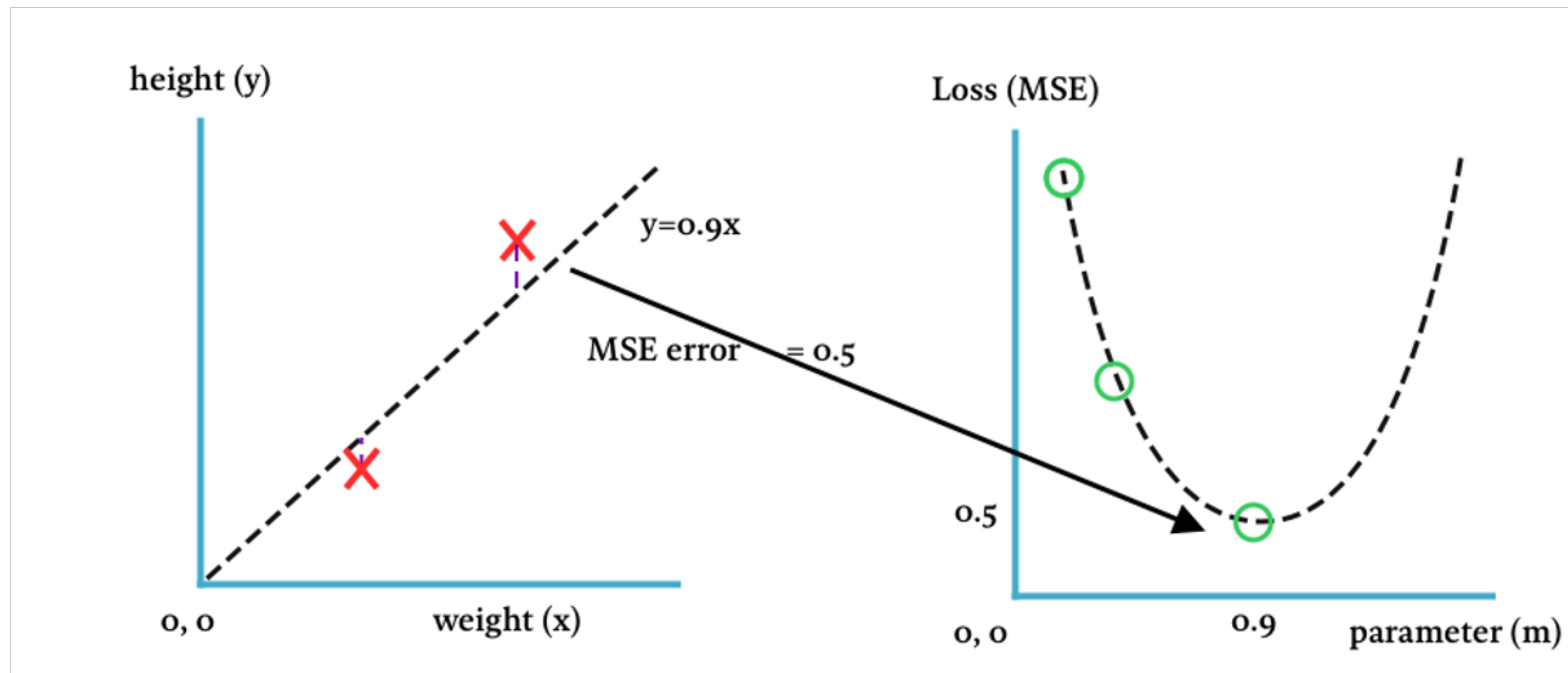
Gradient Descent (Worked out example)

- Why is the curve convex shaped?



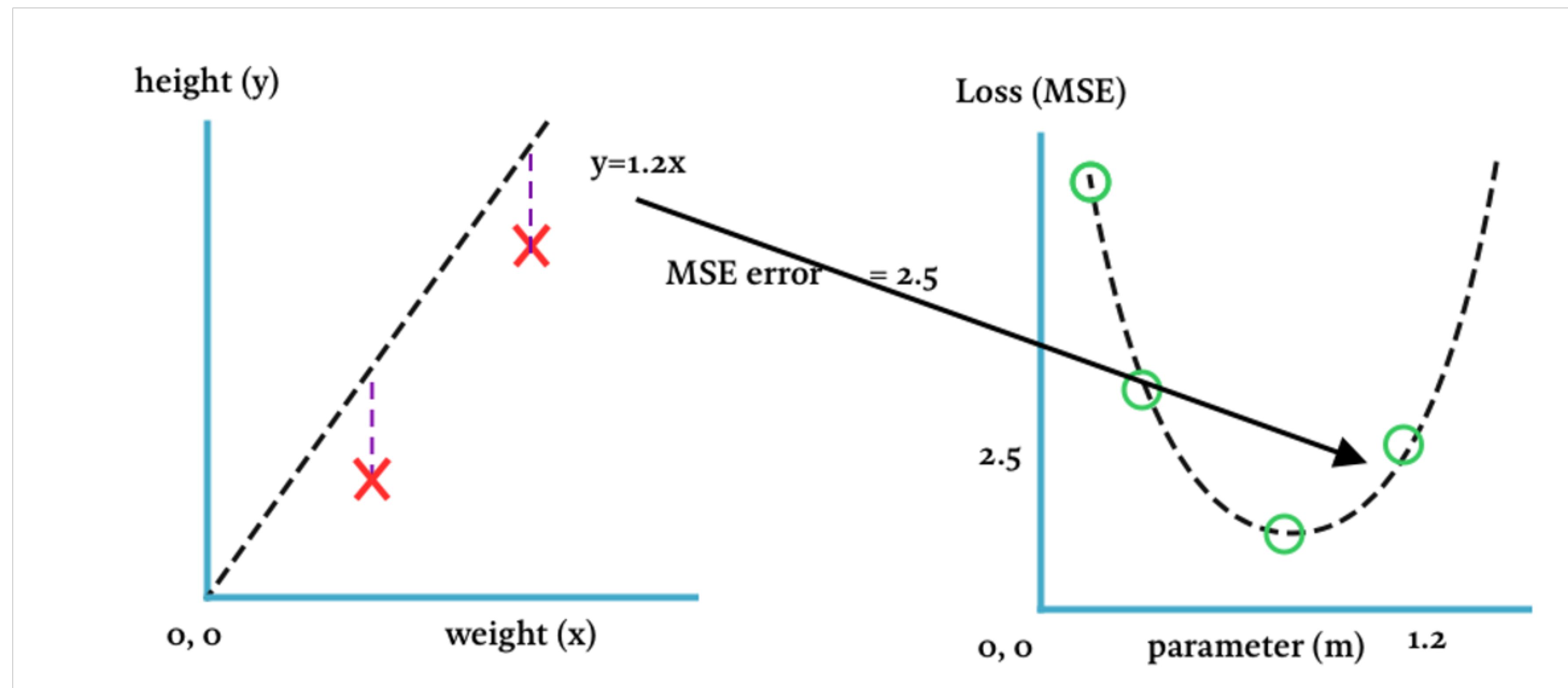
Gradient Descent (Worked out example)

- Why is the curve convex shaped?



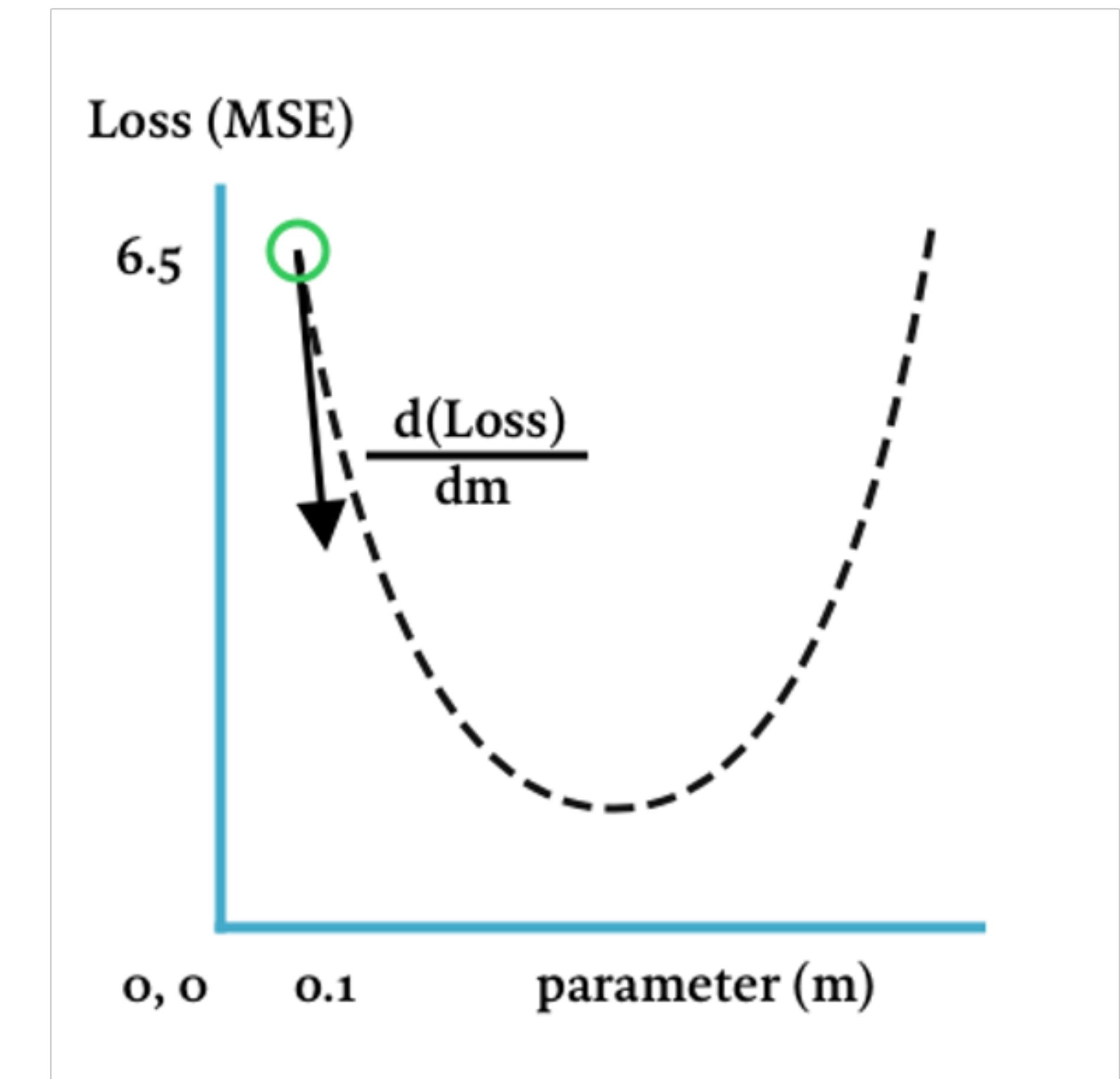
Gradient Descent (Worked out example)

- Why is the curve convex shaped?



Gradient Descent (Worked out example)

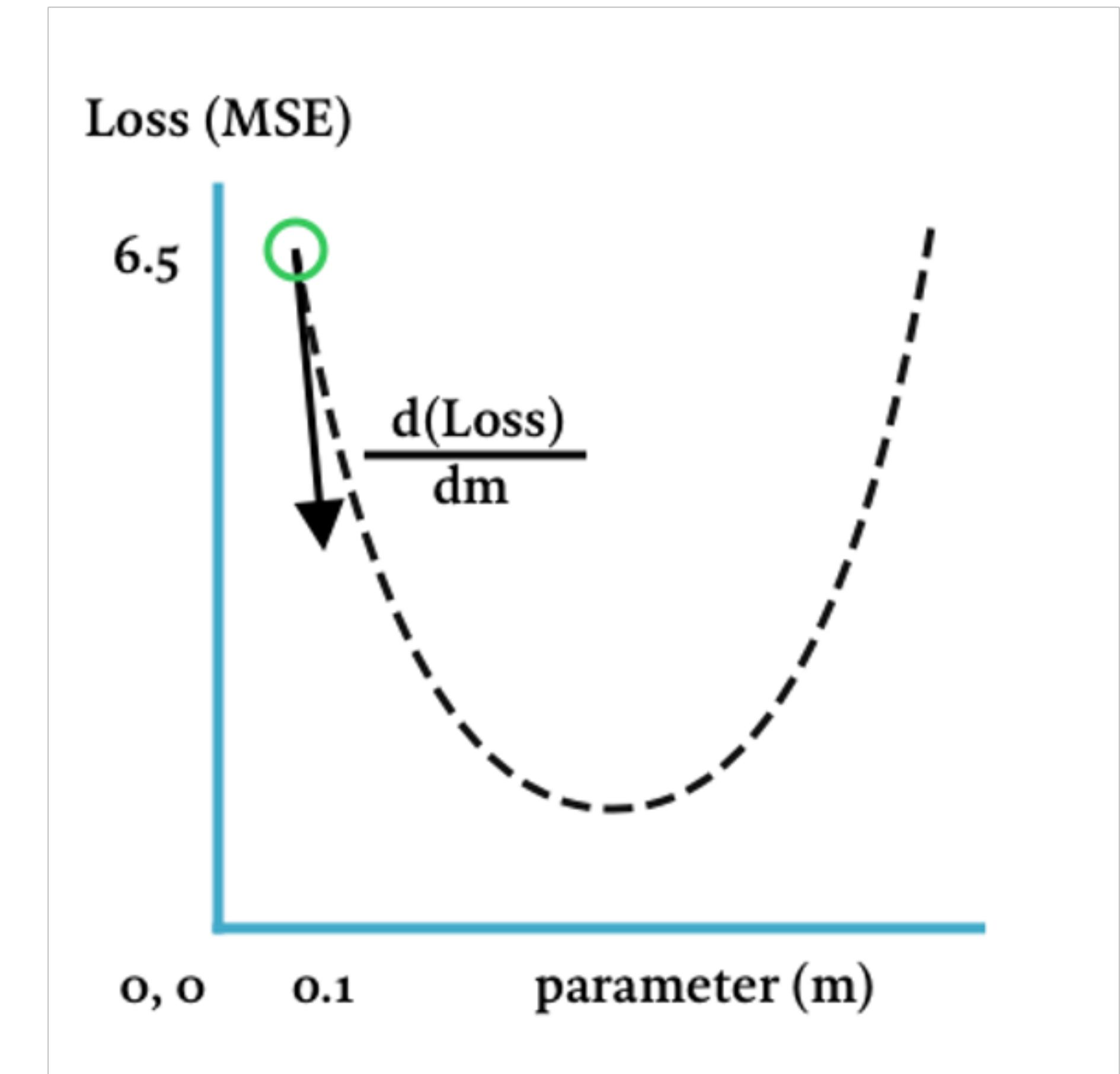
- Step 3: Perform a gradient descent step
 - ▶ Calculate gradient



Gradient Descent (Worked out example)

- Step 3: Perform a gradient descent step
 - ▶ Calculate gradient
 - ▶ Update parameters

$$\mathbf{m}_{\text{new}} = \mathbf{m}_{\text{old}} - \left[-\frac{d(\text{Loss})}{dm} \right] \times \text{step_size}$$



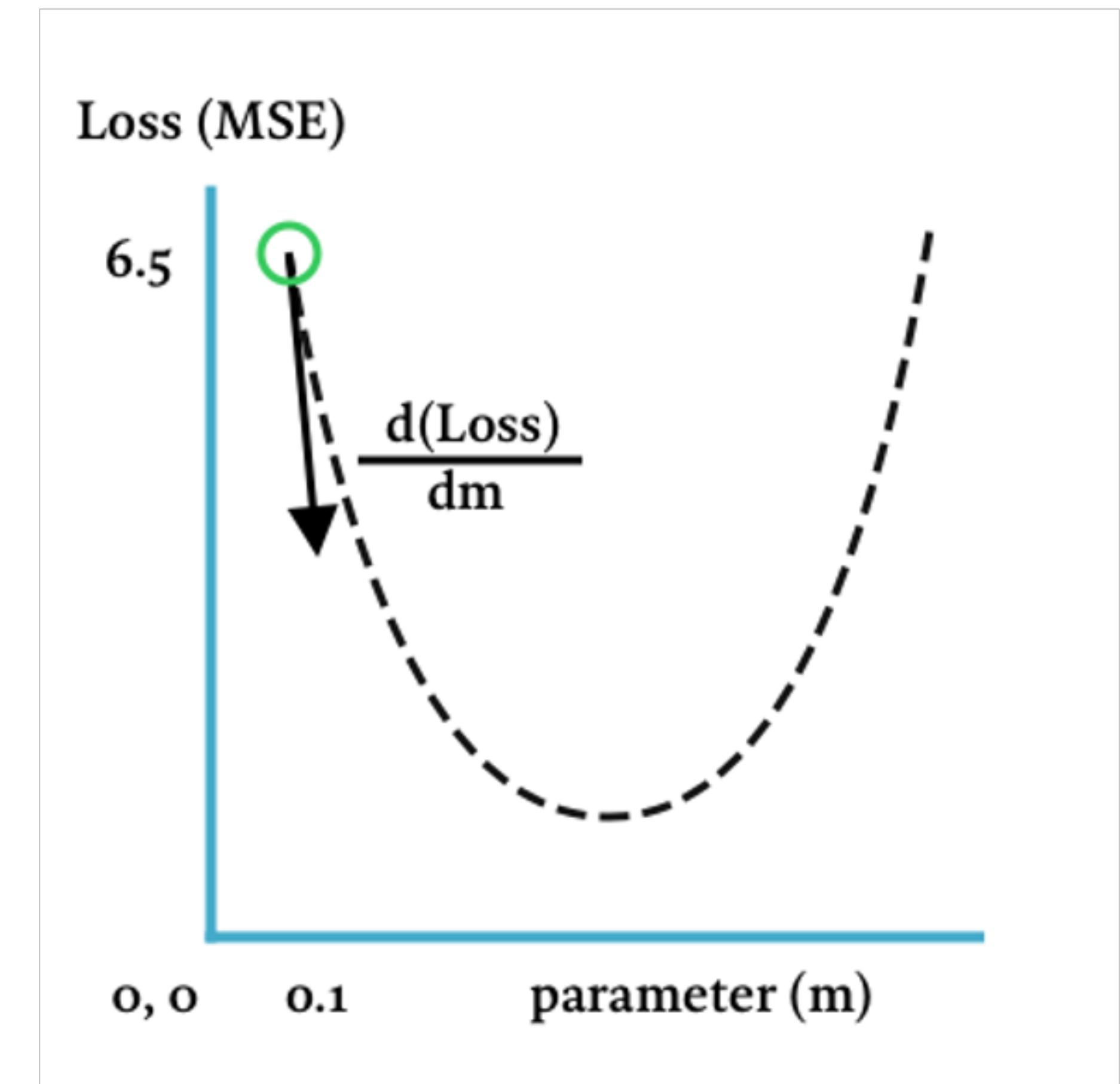
Gradient Descent (Worked out example)

- Step 3: Perform a gradient descent step
 - ▶ Calculate gradient
 - ▶ Update parameters

$$\mathbf{m}_{\text{new}} = \mathbf{m}_{\text{old}} - \left[-\frac{d(\text{Loss})}{dm} \right] \times \text{step_size}$$

$$\mathbf{m}_{\text{new}} = 0.1 - [-2] \times 0.1$$

$$\mathbf{m}_{\text{new}} = 0.3$$



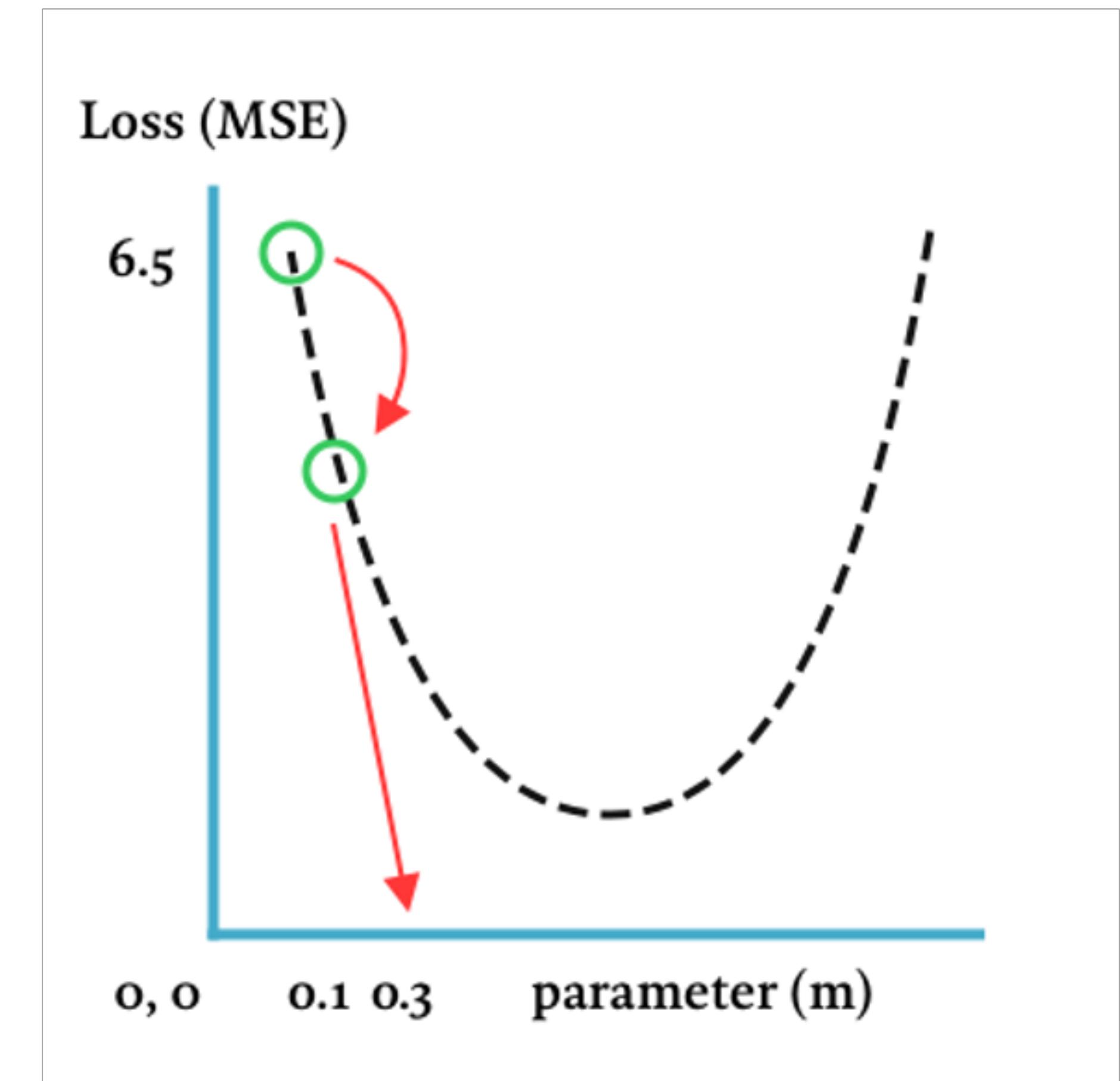
Gradient Descent (Worked out example)

- Step 3: Perform a gradient descent step
 - ▶ Calculate gradient
 - ▶ Update parameters

$$\mathbf{m}_{\text{new}} = \mathbf{m}_{\text{old}} - \left[-\frac{d(\text{Loss})}{dm} \right] \times \text{step_size}$$

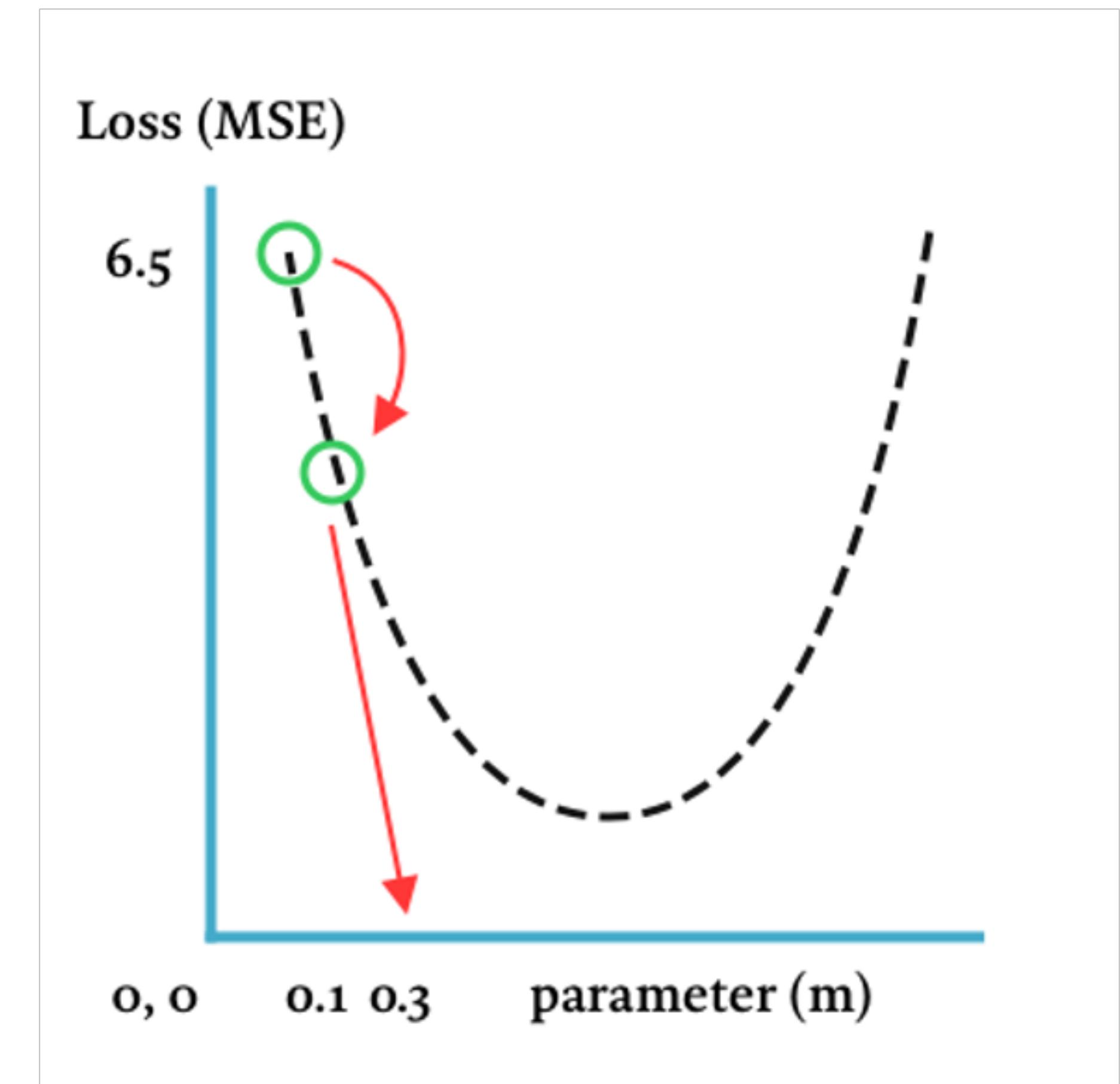
$$\mathbf{m}_{\text{new}} = 0.1 - \left[-2 \right] \times 0.1$$

$$\mathbf{m}_{\text{new}} = 0.3$$



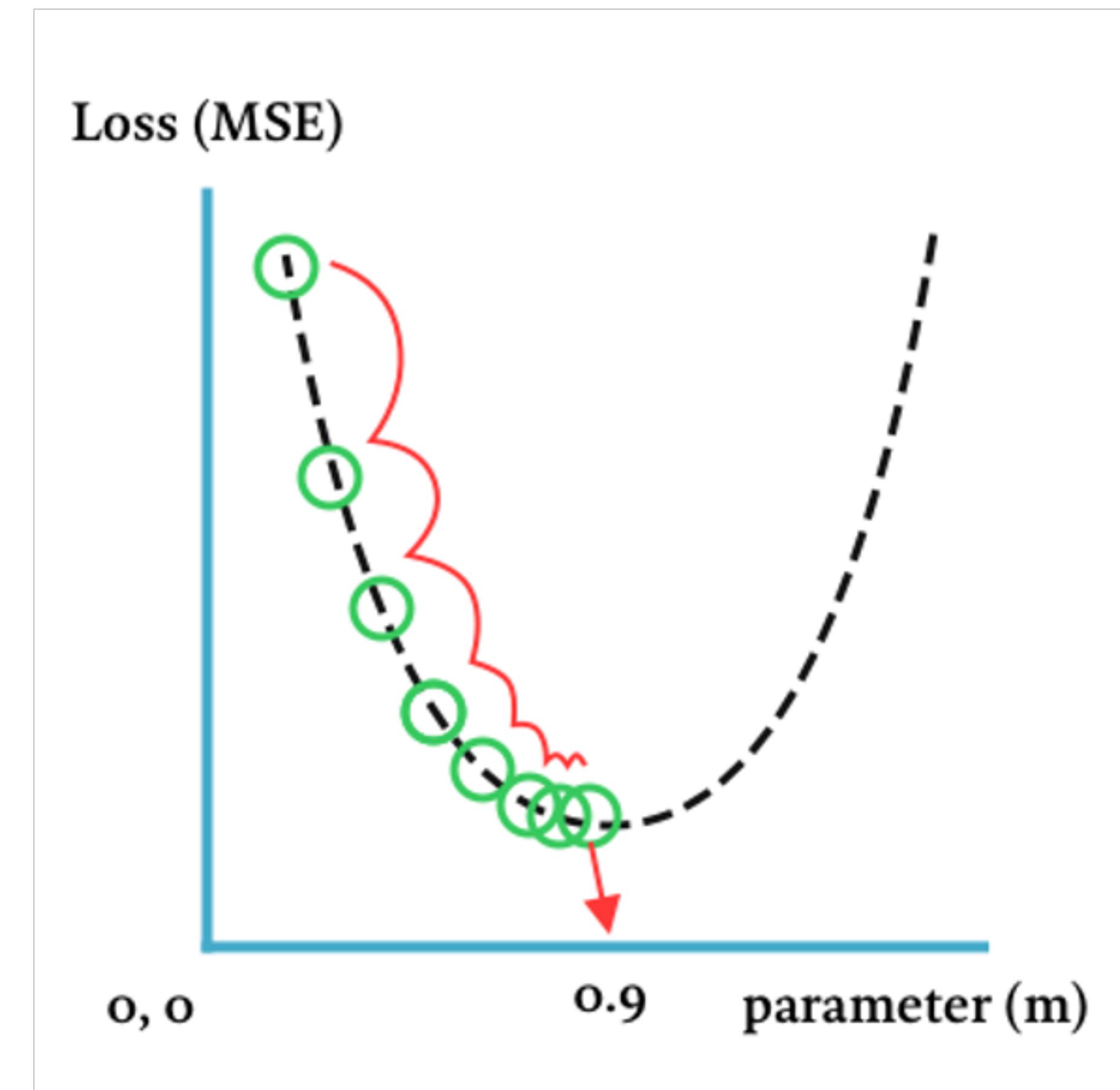
Gradient Descent (Worked out example)

- Step size/ Learning rate
 - ▶ Hyper-parameter
 - ▶ Can't be too large or too small



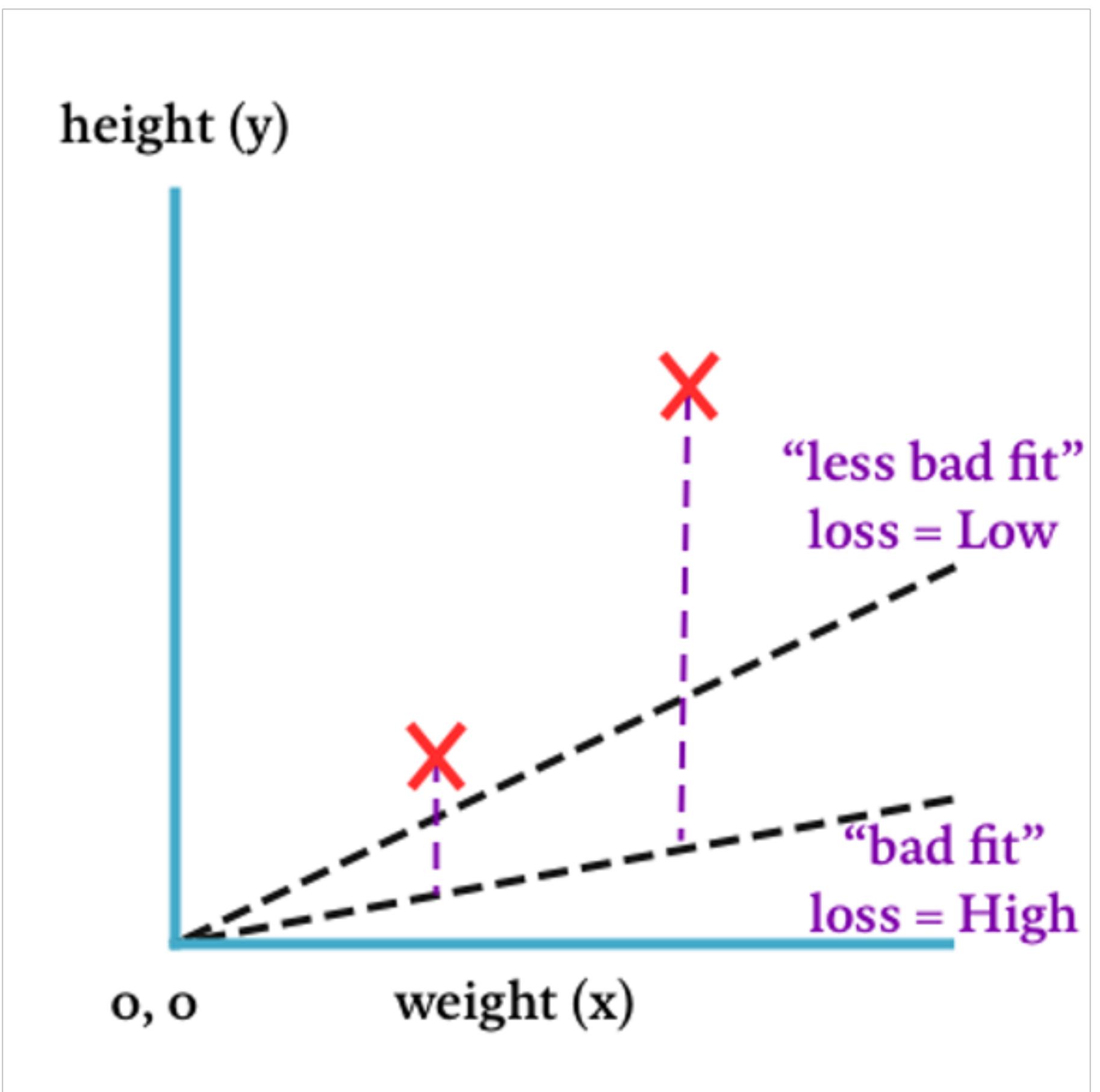
Gradient Descent (Worked out example)

- Step size/ Learning rate trick
 - ▶ Momentum
 - ▶ Faster convergence



Gradient Descent (Worked out example)

- Step 3: Perform a gradient descent step
 - ▶ One step complete

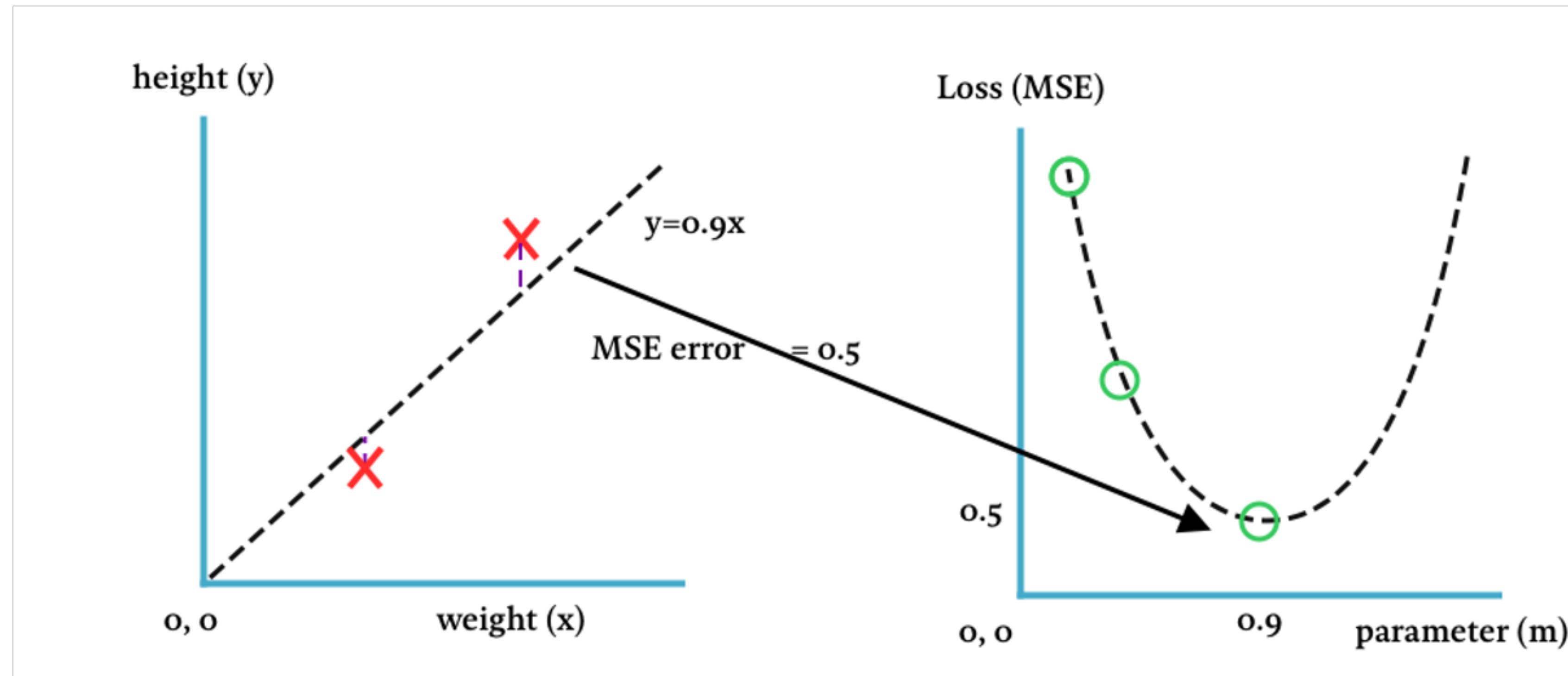


Gradient Descent (Worked out example)

- Step 4: Repeat Step 3 until stopping criteria is met

Gradient Descent (Worked out example)

- Step 4: Repeat Step 3 until stopping criteria is met
 - ▶ Solution found at $m = 0.9$



Gradient Descent (Worked out example)

- Step 4: Repeat Step 3 until **stopping criteria** is met
 - ▶ Gradient = 0
 - ▶ Update < Threshold
 - ▶ Max iterations

Revisit

- Key words
 - ▶ First order
 - ▶ Iterative
 - ▶ Local minimum
 - ▶ Differentiable

Revisit

- Key words
 - ▶ First order: Differentiate loss once
 - ▶ Iterative
 - ▶ Local minimum
 - ▶ Differentiable

Revisit

- Key words
 - ▶ First order: Differentiate loss once
 - ▶ Iterative: Optimal value by repeated updates
 - ▶ Local minimum
 - ▶ Differentiable

Revisit

- Key words
 - ▶ First order: Differentiate loss once
 - ▶ Iterative: Optimal value by repeated updates
 - ▶ Local minimum: Each iteration, lowest point in neighborhood
 - ▶ Differentiable

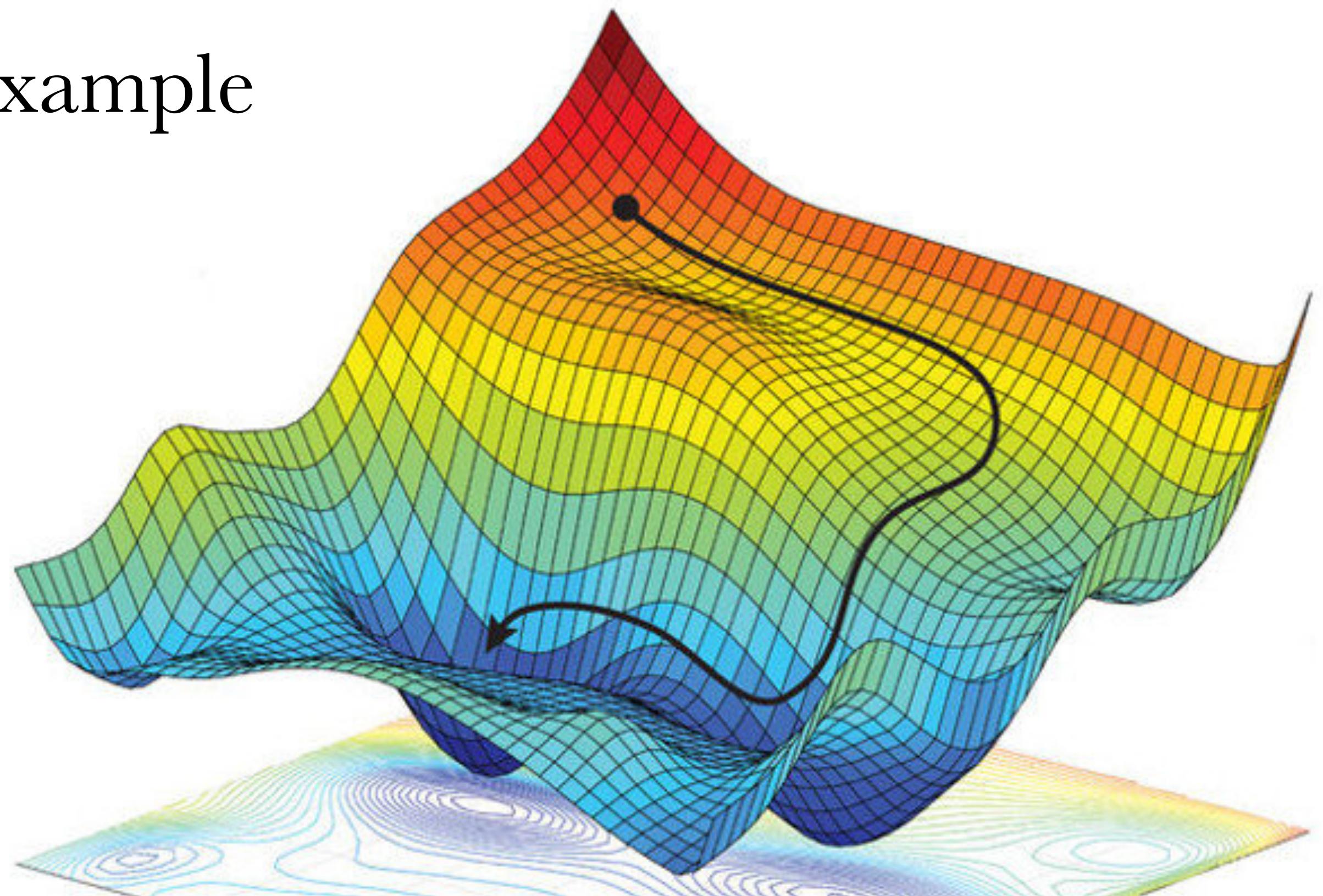
Revisit

- Key words
 - ▶ First order: Differentiate loss once
 - ▶ Iterative: Optimal value by repeated updates
 - ▶ Local minimum: Each iteration, lowest point in neighborhood
 - ▶ Differentiable: Loss function allowed for differentiation



Gradient Descent Analogy

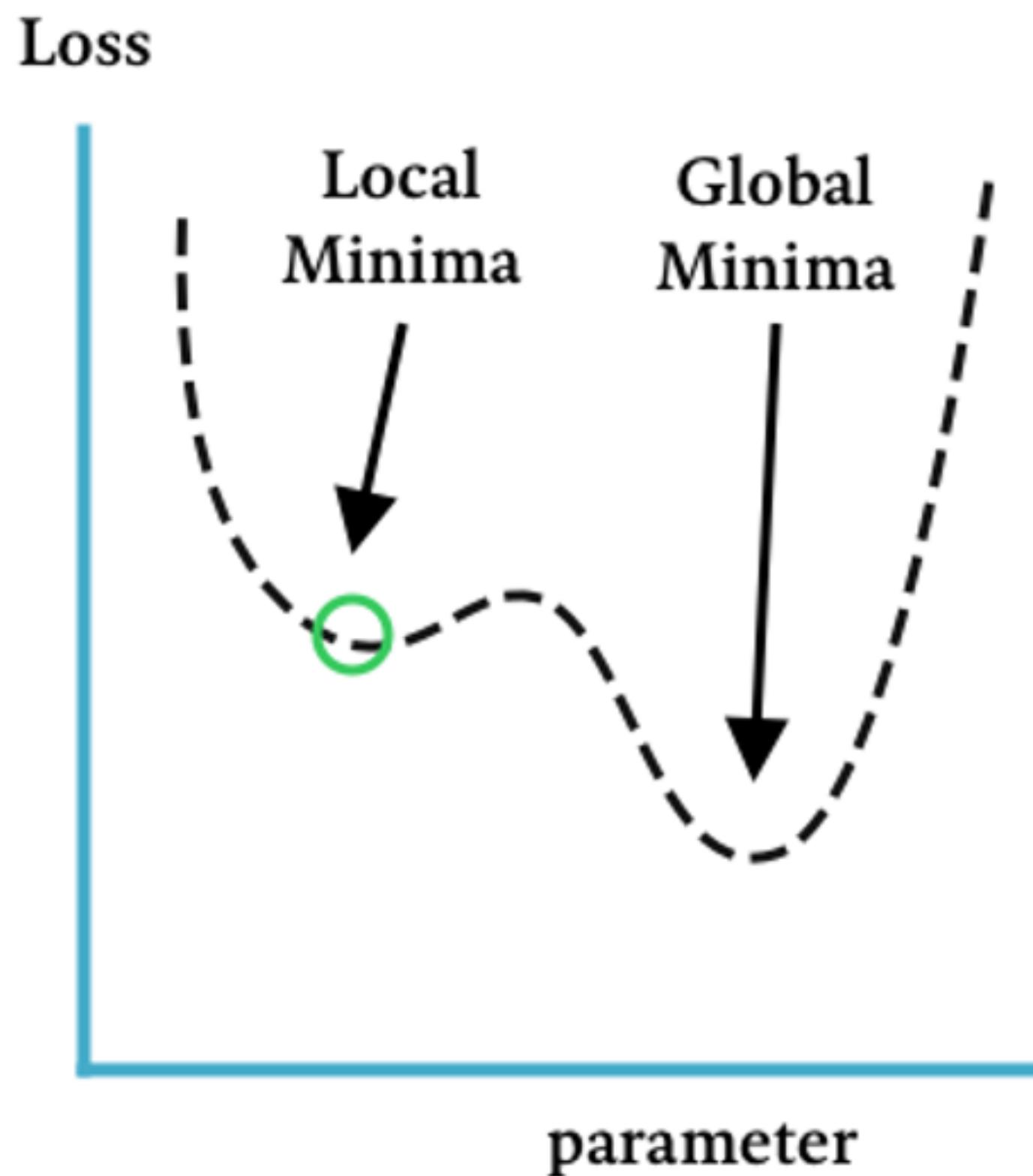
- Think of a multi-dimensional example



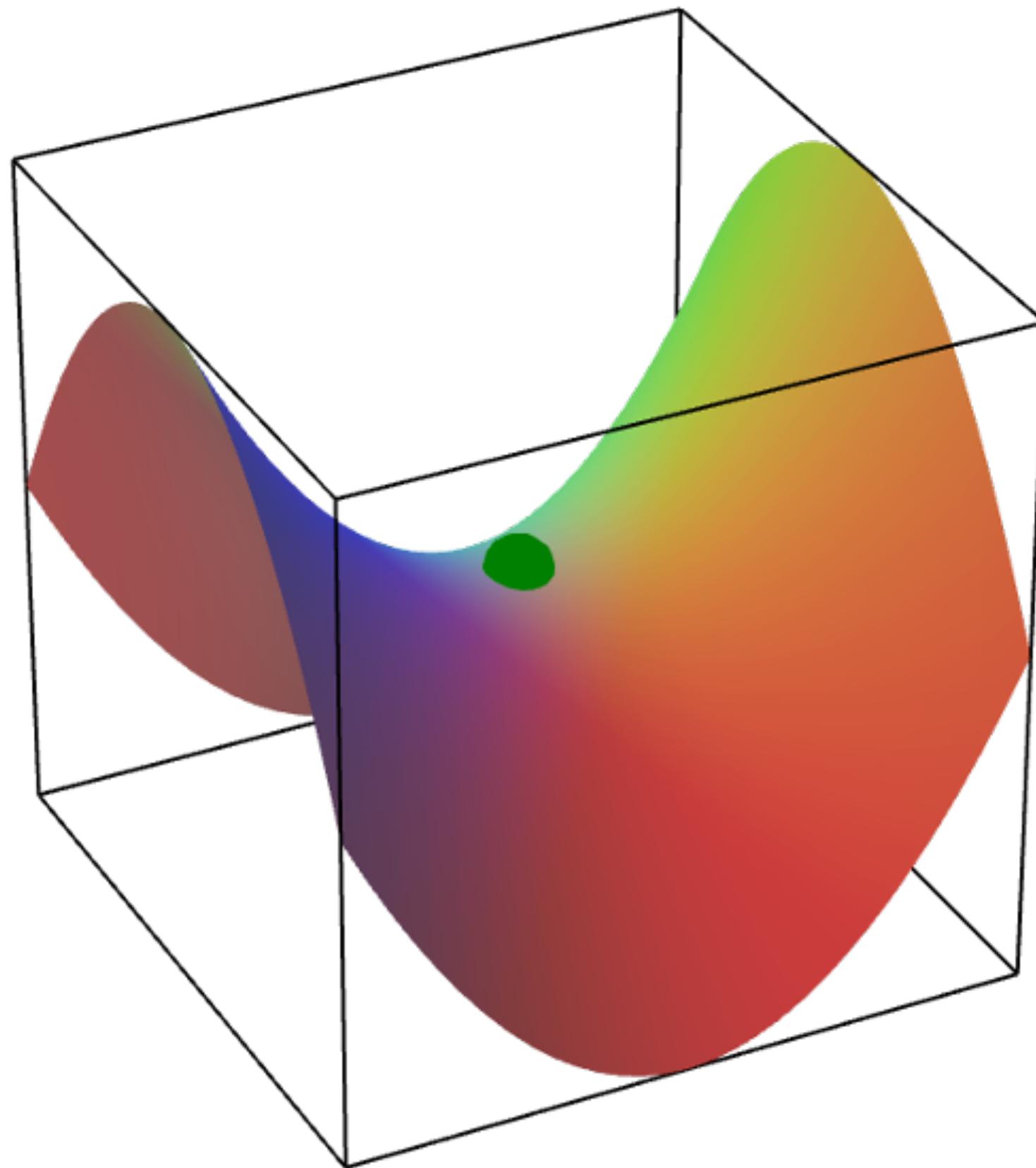
Problems with Gradient Descent

Problems with gradient descent

- Stuck in local minima

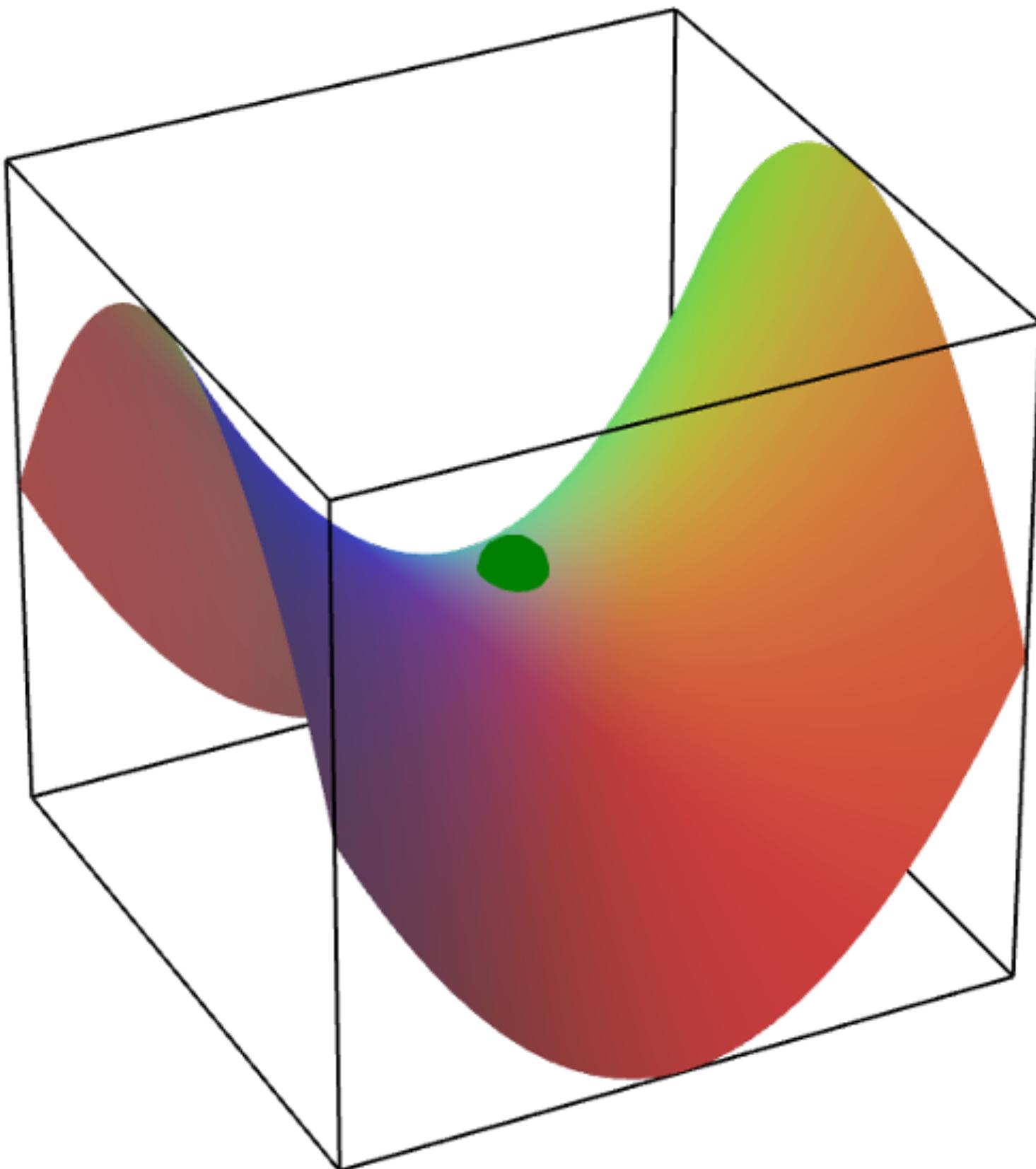


Problems with gradient descent



- Stuck in local minima
- Stuck in a saddle point

Problems with gradient descent



- Stuck in local minima
- Stuck in a saddle point
 - ▶ Sub optimal solutions
 - ▶ No incentive to update parameter

More Problems with gradient descent

- When **scale** and **dimensionality** of data is high

More Problems with gradient descent

- When **scale** and **dimensionality** of data is high
 - ▶ Model (10,000 params) to classify images
 - ▶ 100,000 images
 - ▶ Each image 1 Megapixel= 1,000,000 features

More Problems with gradient descent

- When **scale** and **dimensionality** of data is high
 - ▶ Model (10,000 params) to classify images
 - ▶ 100,000 images
 - ▶ Each image 1 Megapixel= 1,000,000 features
 - ▶ At each step, $\sim(1,000,000,000,000 + 10,000)$ computations to obtain gradients?

More Problems with gradient descent

- When **scale** and **dimensionality** of data is high
- How to perform gradient descent here?

Stochastic Gradient Descent

Stochastic Gradient Descent

- Stochastic approximation of the true gradient

Stochastic Gradient Descent

- Stochastic approximation of the true gradient
 - ▶ Use less data
 - ▶ Take approximate steps
 - ▶ Take them faster!!

Stochastic Gradient Descent

- Use less data
 - ▶ Randomly sample a **batch**

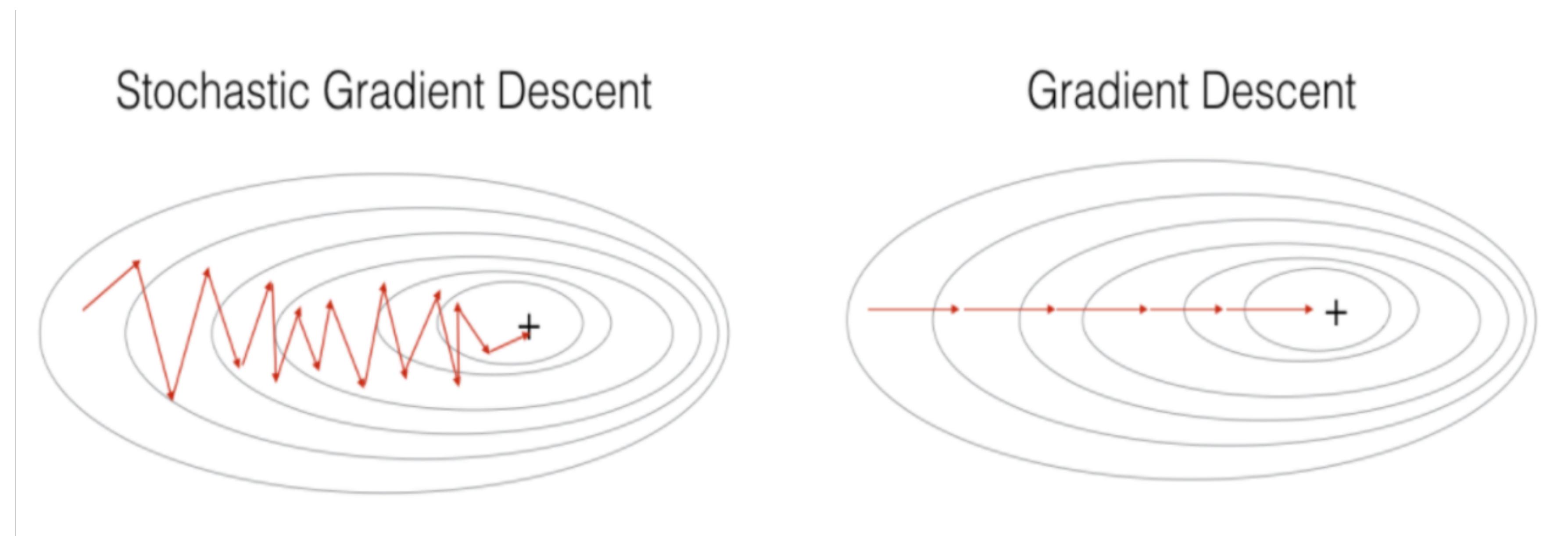


Stochastic Gradient Descent

- Use less data
 - ▶ Randomly sample a **batch**
 - ▶ Batch Gradient Descent (true gradient)
 - ▶ SGD/ Mini-Batch Gradient Descent (sample size ≥ 1)

Stochastic Gradient Descent

- Take approximate steps
 - ▶ 2D view of going down the valley



Stochastic Gradient Descent

- Take approximate steps
 - ▶ 2D view of going down the valley
 - ▶ Smaller variance = better estimate

Stochastic Gradient Descent

- Take steps faster
 - ▶ Mini-batch enables parallelism
 - ▶ Enables use of a GPU

Stochastic Gradient Descent

- SGD and variance
 - ▶ High variance near optimal solution (example)
 - ▶ Early stopping

Stochastic Gradient Descent

- SGD in Neural Networks
 - ▶ Backpropagation = calculate a single stochastic gradient

Stochastic Gradient Descent

- SGD in Neural Networks
 - ▶ Backpropagation = calculate a single stochastic gradient
 - ▶ Implementation challenges:
 - Step-size?
 - Mini-batch size?

Recap

Recap

- Optimization
 - ▶ Maximize or minimize an objective function to find best fit parameters

Recap

- Gradient Descent
 - ▶ Iterative algorithm that uses gradients to solve optimization

Recap

- Stochastic Gradient Descent
 - ▶ Uses random sampling to apply gradient descent on data with high scale and dimensionality

Thank You!

Bibek Poudel
bpoudel@memphis.edu

But wait... there's more...

I DON'T USUALLY MINIMIZE
FUNCTIONS



BUT WHEN I DO, I CHOOSE
STOCHASTIC GRADIENT DESCENT

.net

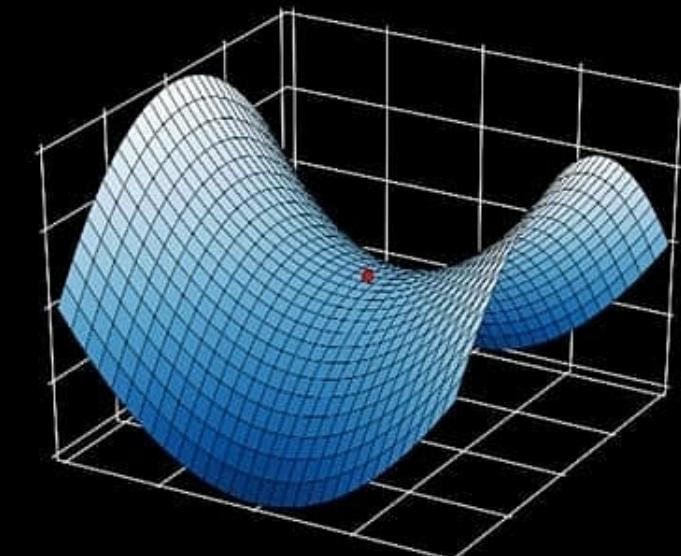
What others see



Potato chip



What i see



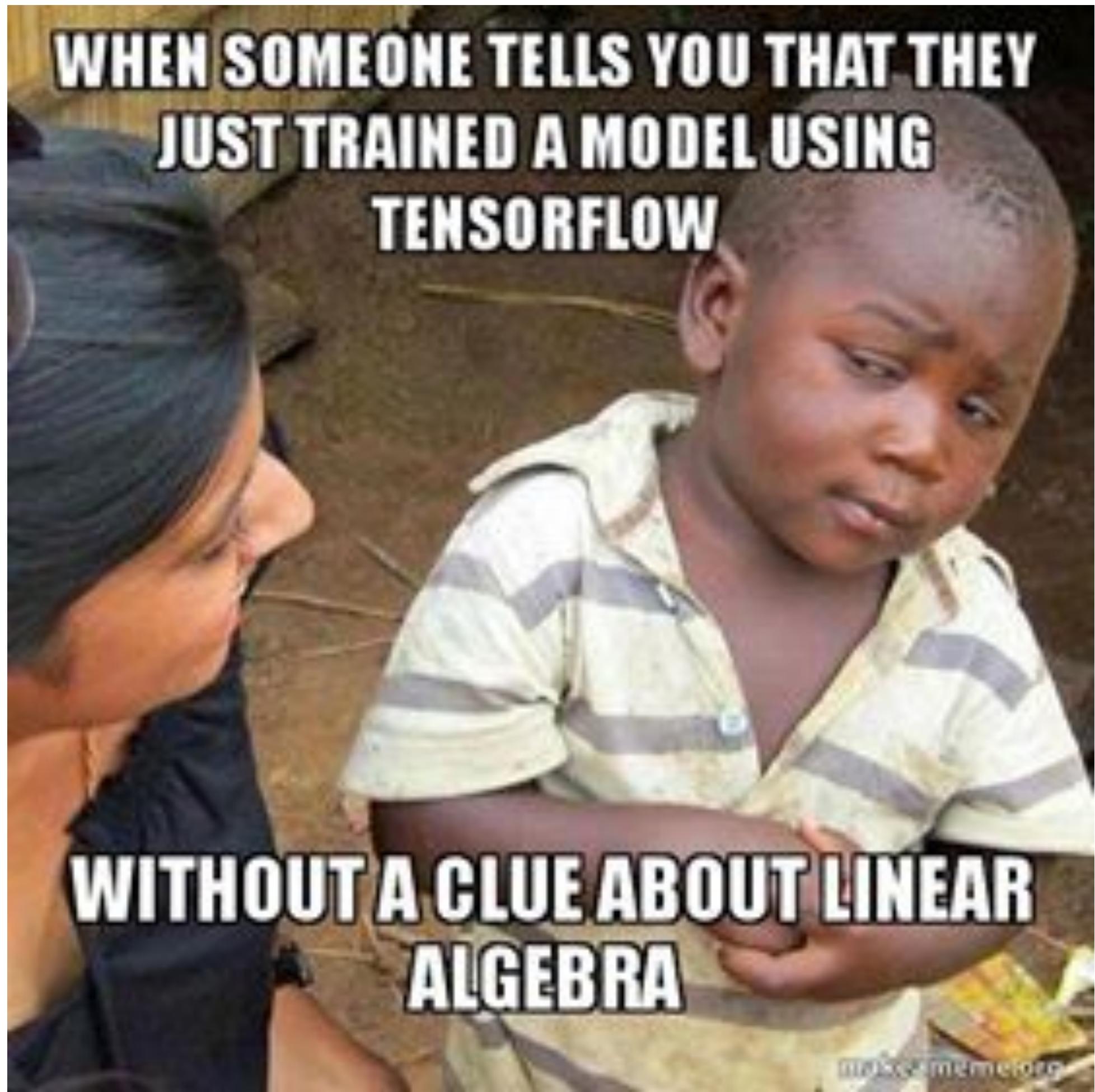
Gradient descent

Gradient descent is an iterative optimization algorithm for finding the **minimum** of a function.

$$x_{i+1} = x_i - \alpha \nabla f(x_i)$$

Diagram illustrating the gradient descent update rule:

- next position*: The final position after the update.
- opposite direction*: The direction of the gradient vector, pointing away from the minimum.
- gradient at current position*: The gradient vector at the current position x_i .
- current position*: The initial position before the update.
- step size (learning rate)*: The scalar factor α used to scale the gradient vector.





Global
minimum

Stochastic
gradient
descent