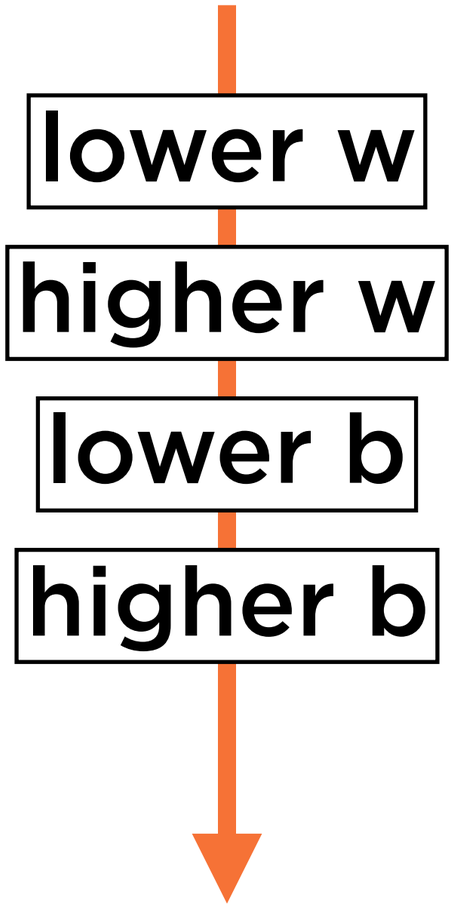
Improving the Algorithm with Gradient Descent

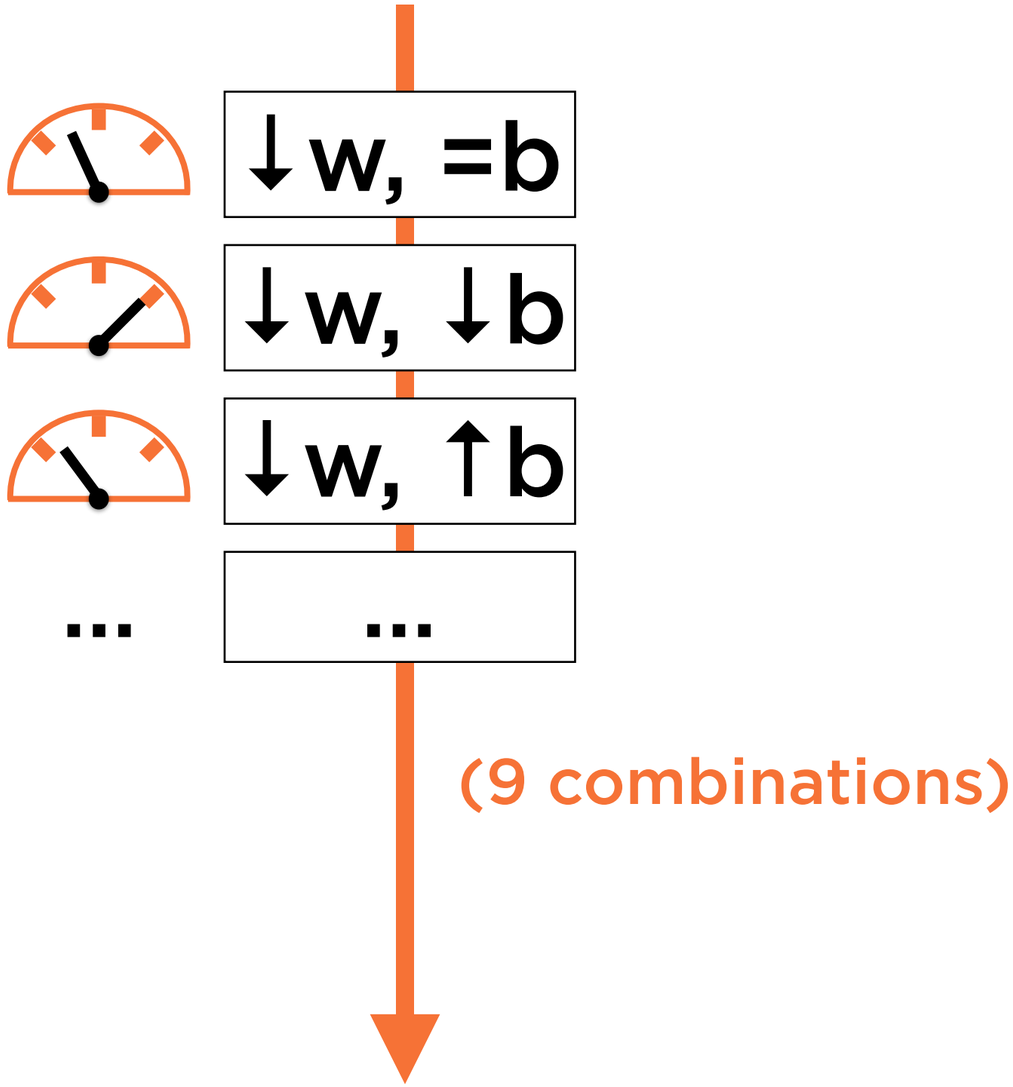
Paolo Perrotta
FREELANCE DEVELOPER
@nusco

Trial and Error



Changing w might increase the loss caused by b, and the other way around.

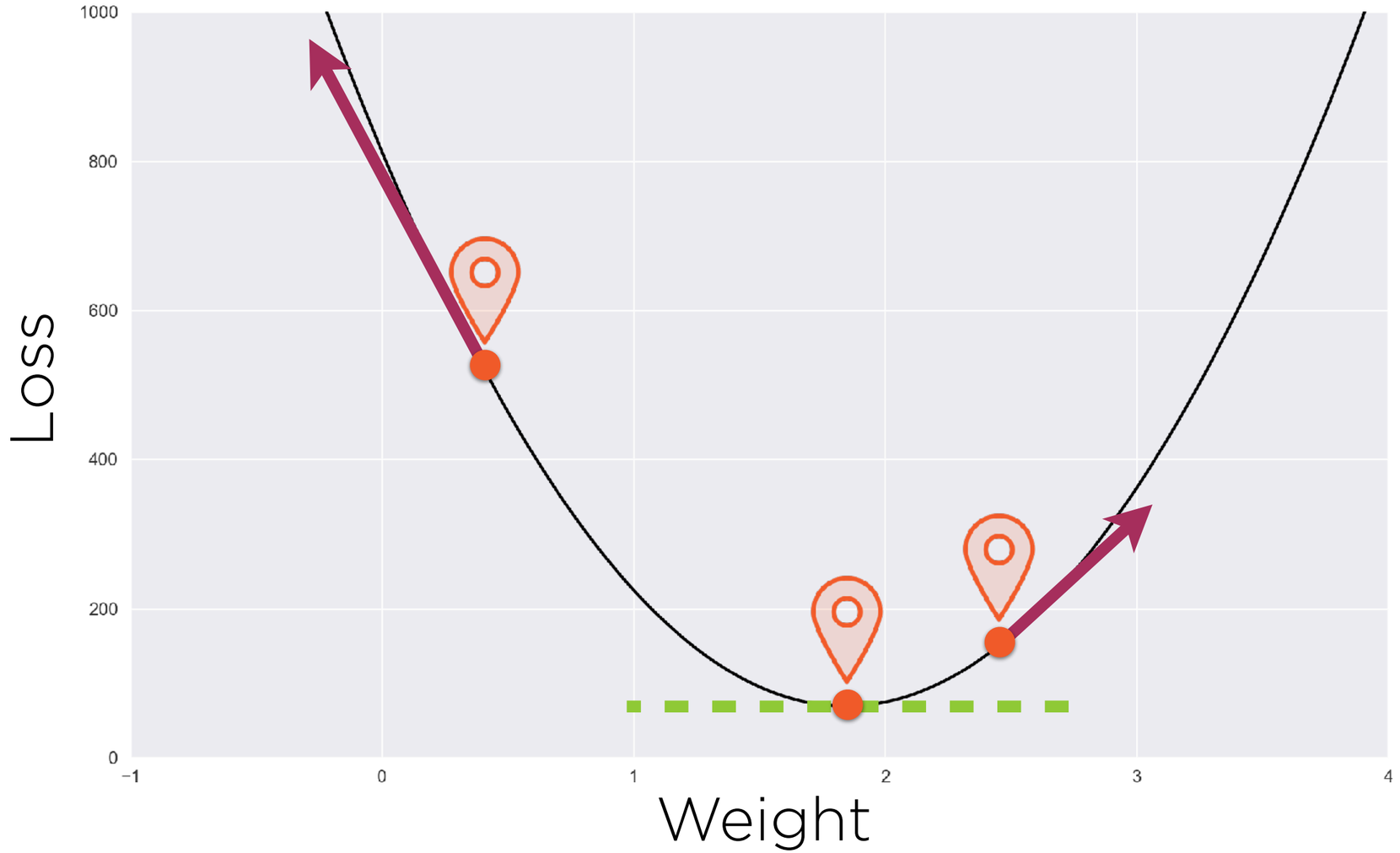
Trying Every Combination of Parameters



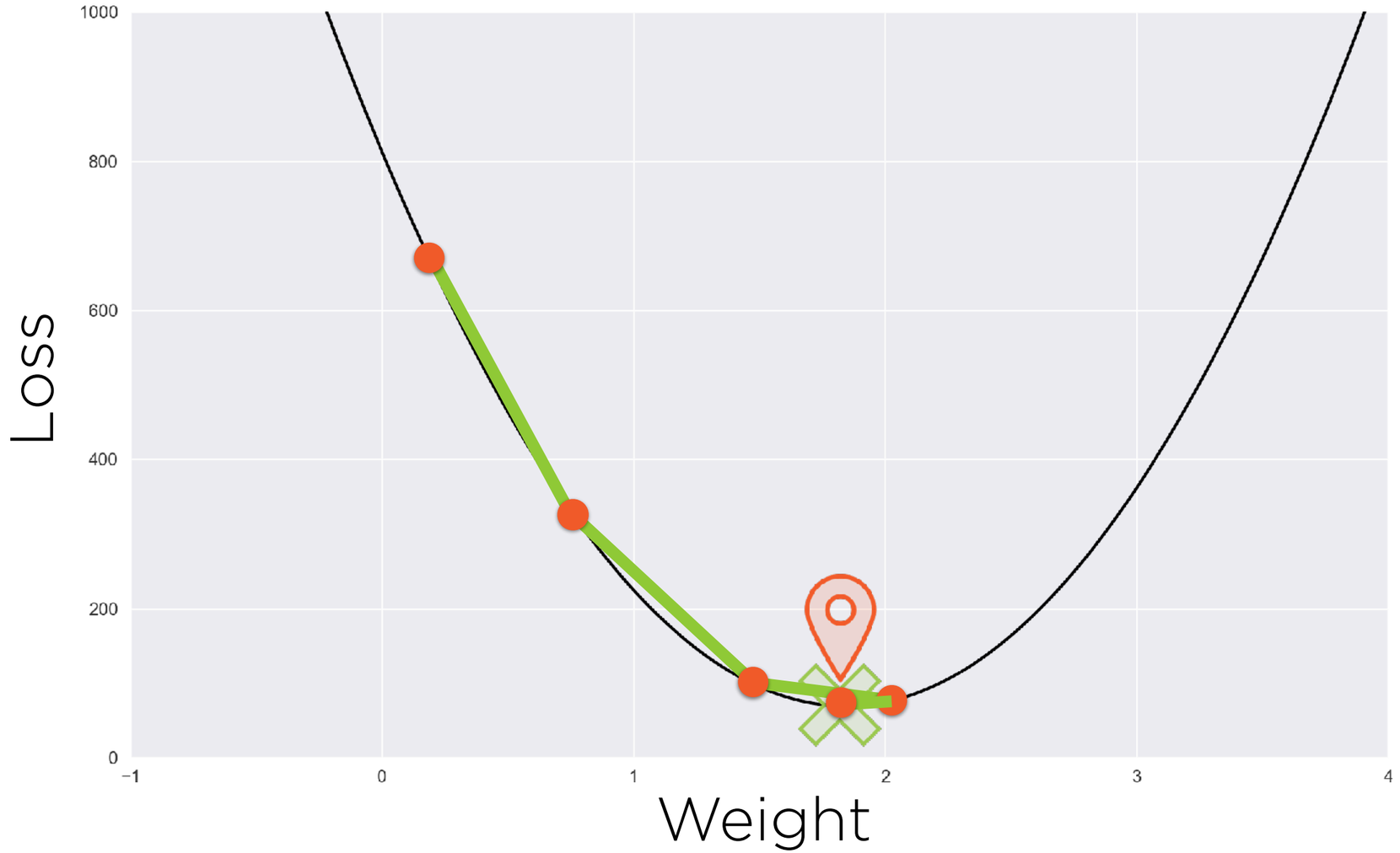
Trying Every Combination of Parameters

Parameters	Combinations
2	9
3	27
4	64
1000	1 billion
1 million	1000000000000000000
...	...

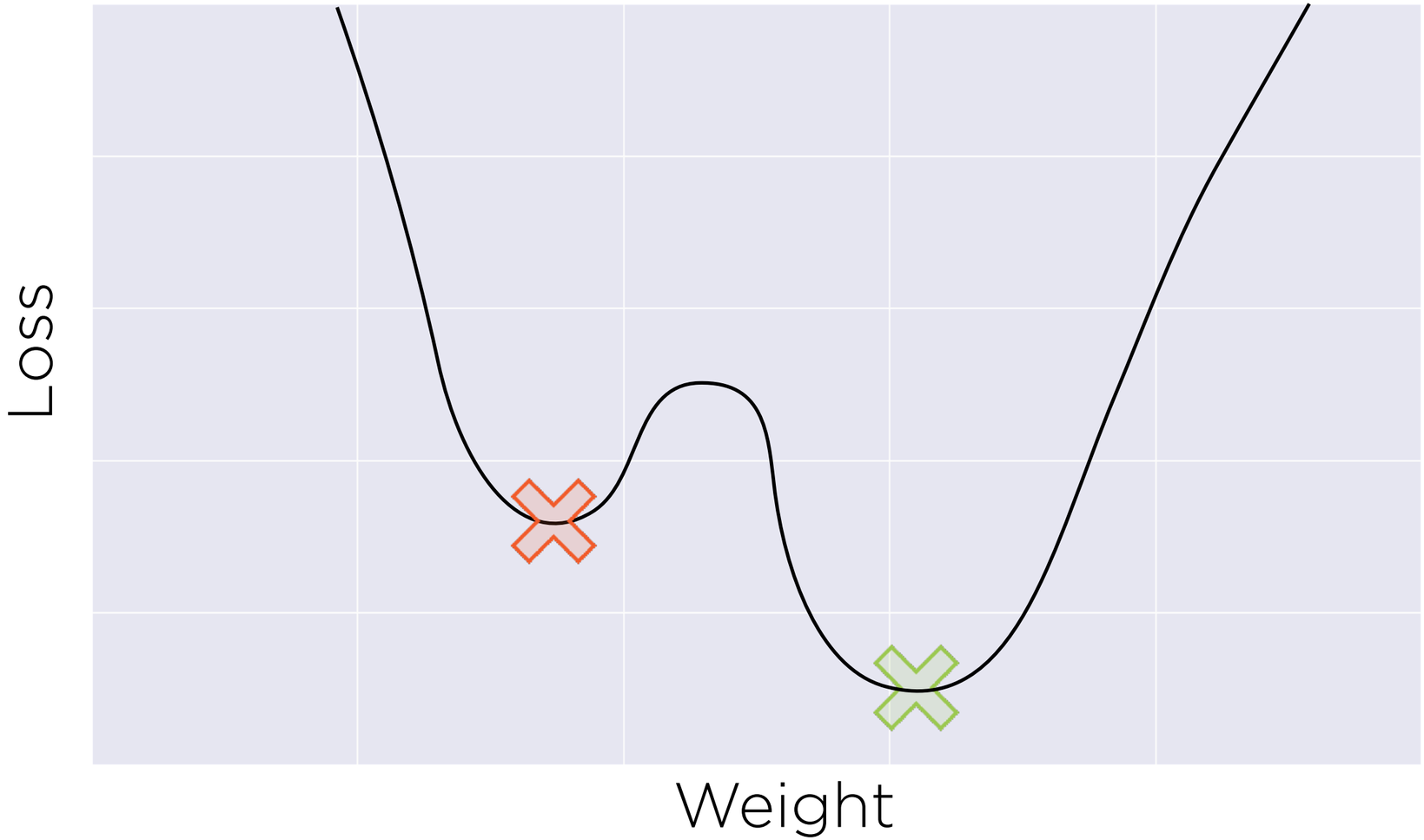
The Gradient



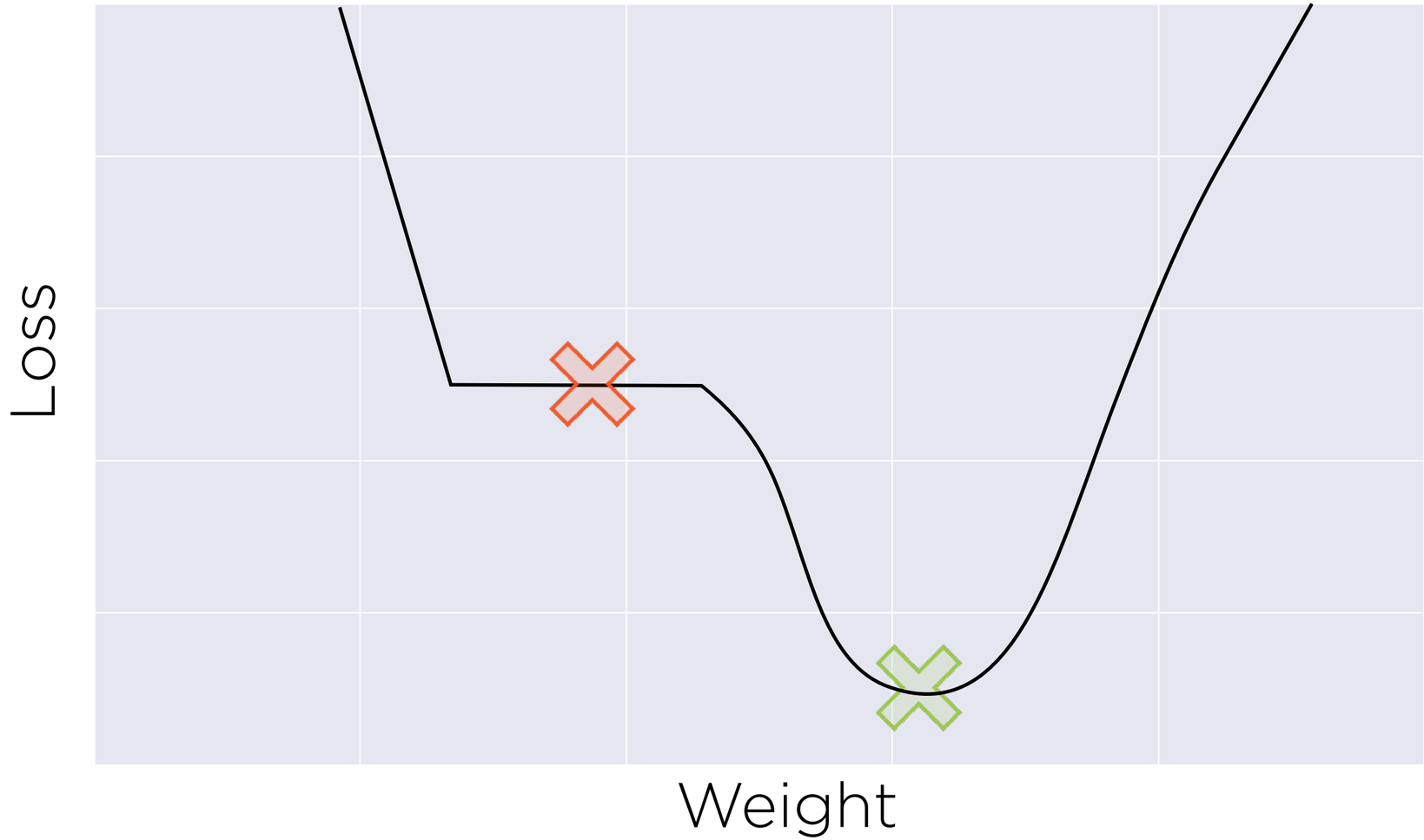
Gradient Descent



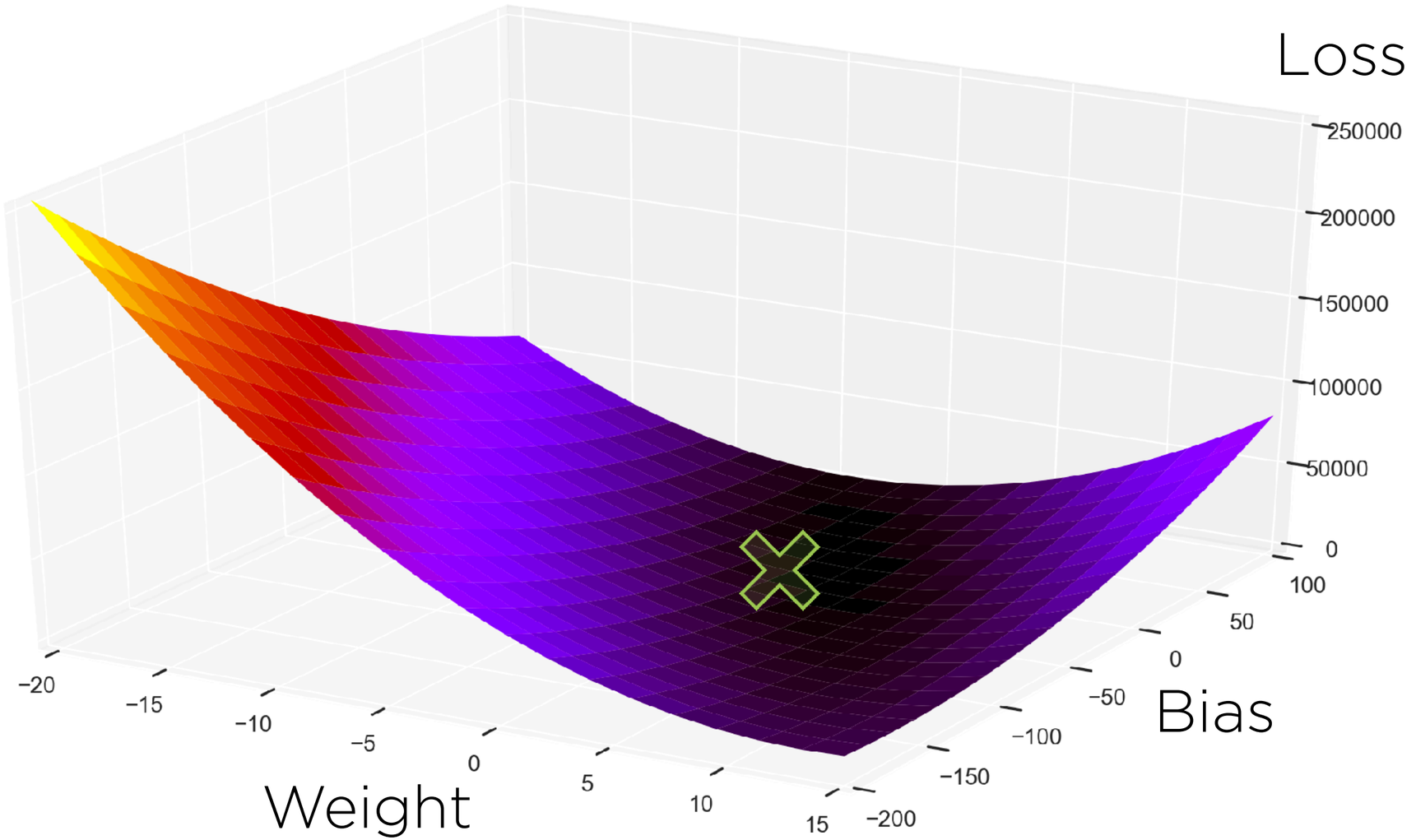
Limitations of Gradient Descent



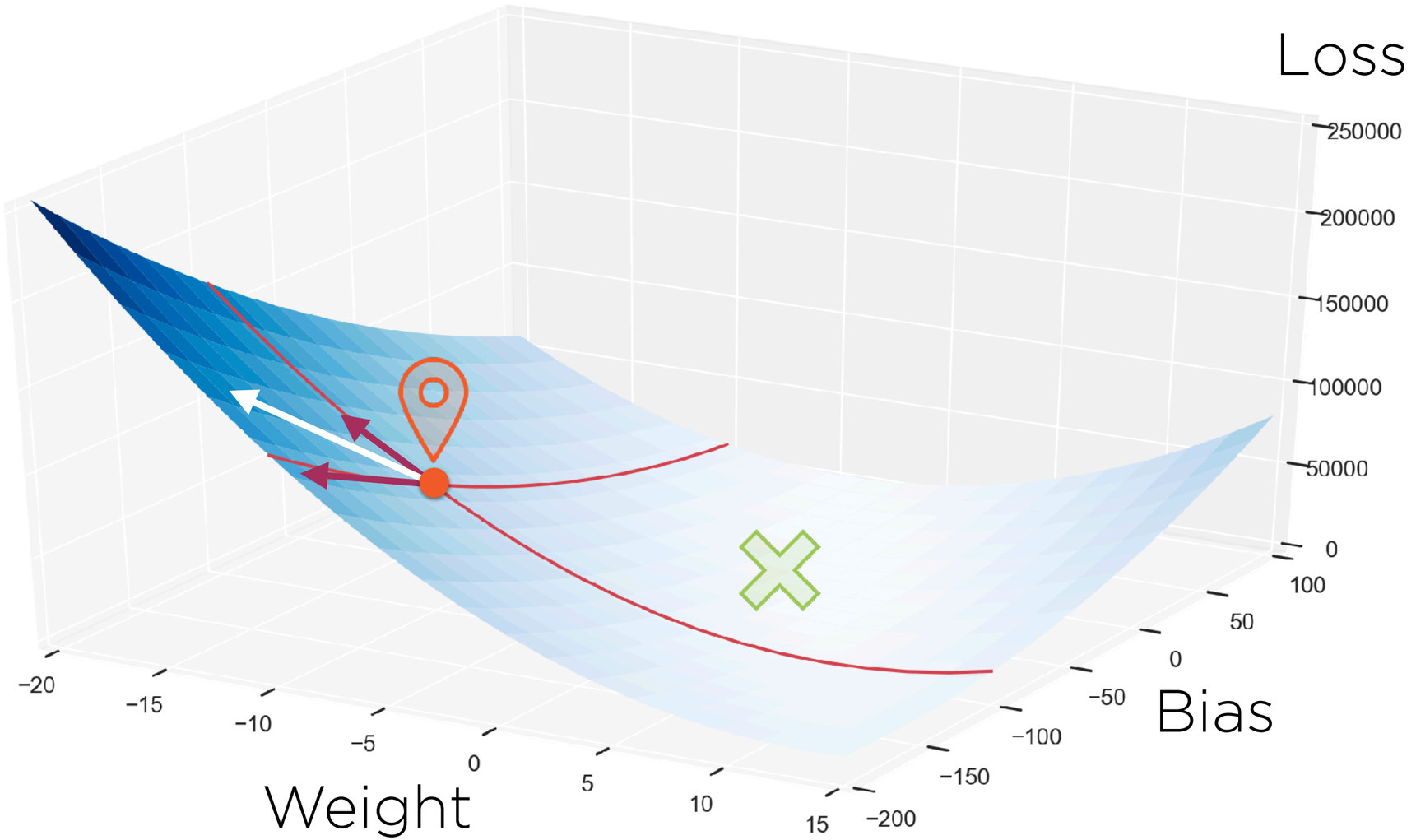
Limitations of Gradient Descent



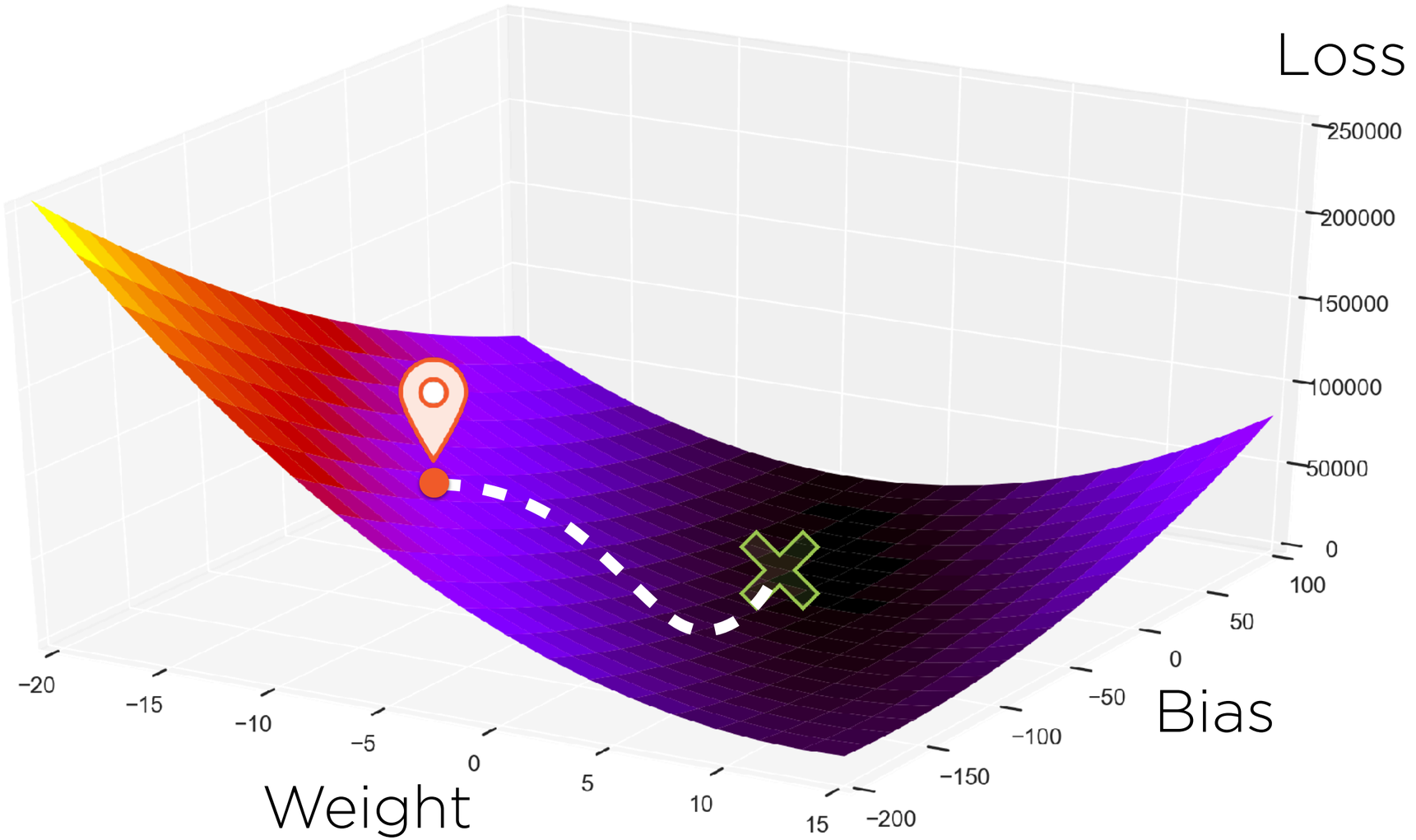
The Loss Curve in Three Dimensions



Gradient Descent in Three Dimensions



Gradient Descent in Three Dimensions



Calculating the Gradient

The Loss and Its Gradient

$$L = \frac{1}{m}\sum((wx + b) - y)^2$$

$$\frac{\partial L}{\partial w} = \frac{1}{m}\sum 2x((wx + b) - y)$$

$$\frac{\partial L}{\partial b} = \frac{1}{m}\sum 2((wx + b) - y)$$

Summary

We learned the limitations of our training algorithm

We replaced it with gradient descent

- Start with random parameters
- Calculate the gradient of the loss
- Take a step in the opposite direction
- Repeat for a while

Gradient descent works well for our model and loss