[320] Optimization and Gradient Descent

Tyler Caraza-Harter

Optimization Problems

minimize or maximize something

1. find the x value that **minimizes** the y, when y=f(x)

2. find the fit line coeficients (slope and intercept) that **minimize** the average squared differences between the data and the line

3. find the weights on edges between neurons to **minimize** the mistakes made by the neural network

Techniques

1 find the x value that **minimizes** the y, when y=f(x)

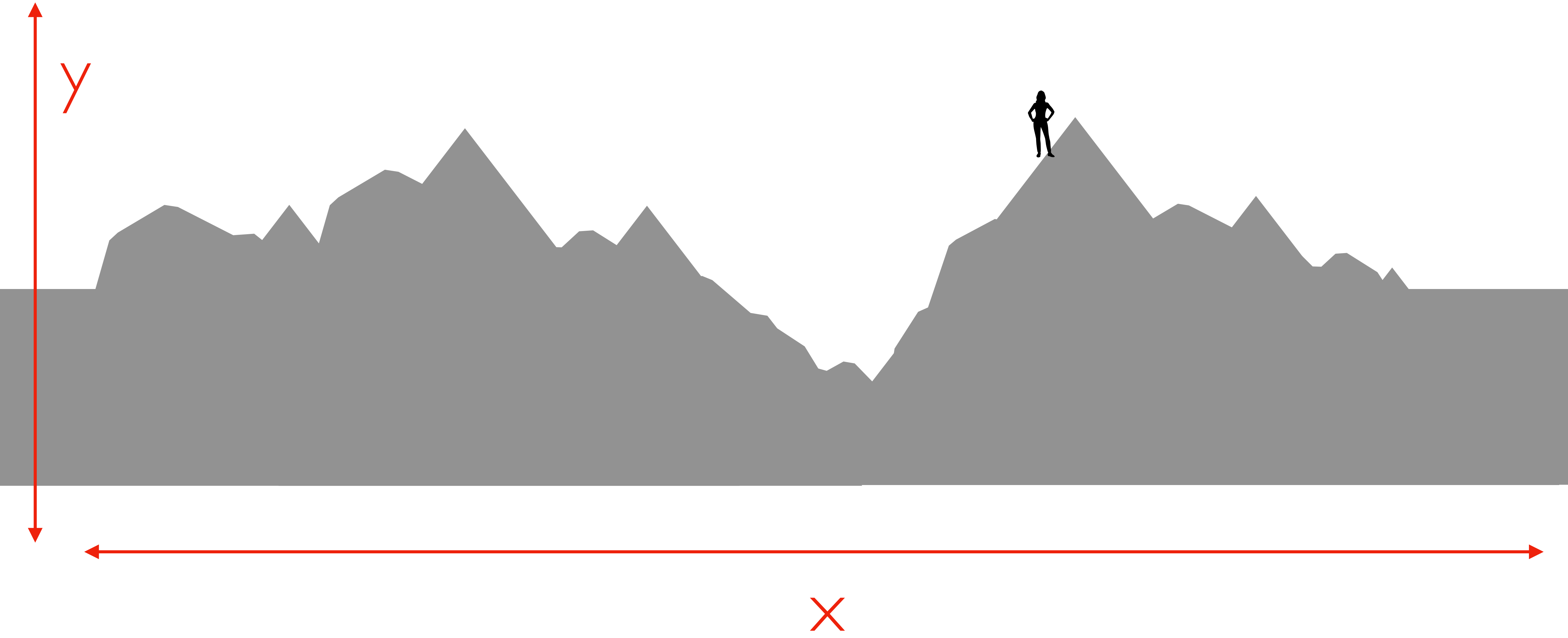
Calculus: find derivative of continuous function f, set to zero, evaluate x solutions

Compute: loop over lots of x values (-5, -4.9, -4.8, ..., 4.8, 4.9, 5)

Compute: gradient descent (keep tweaking x based on gradient, searching for best)

Gradient Descent

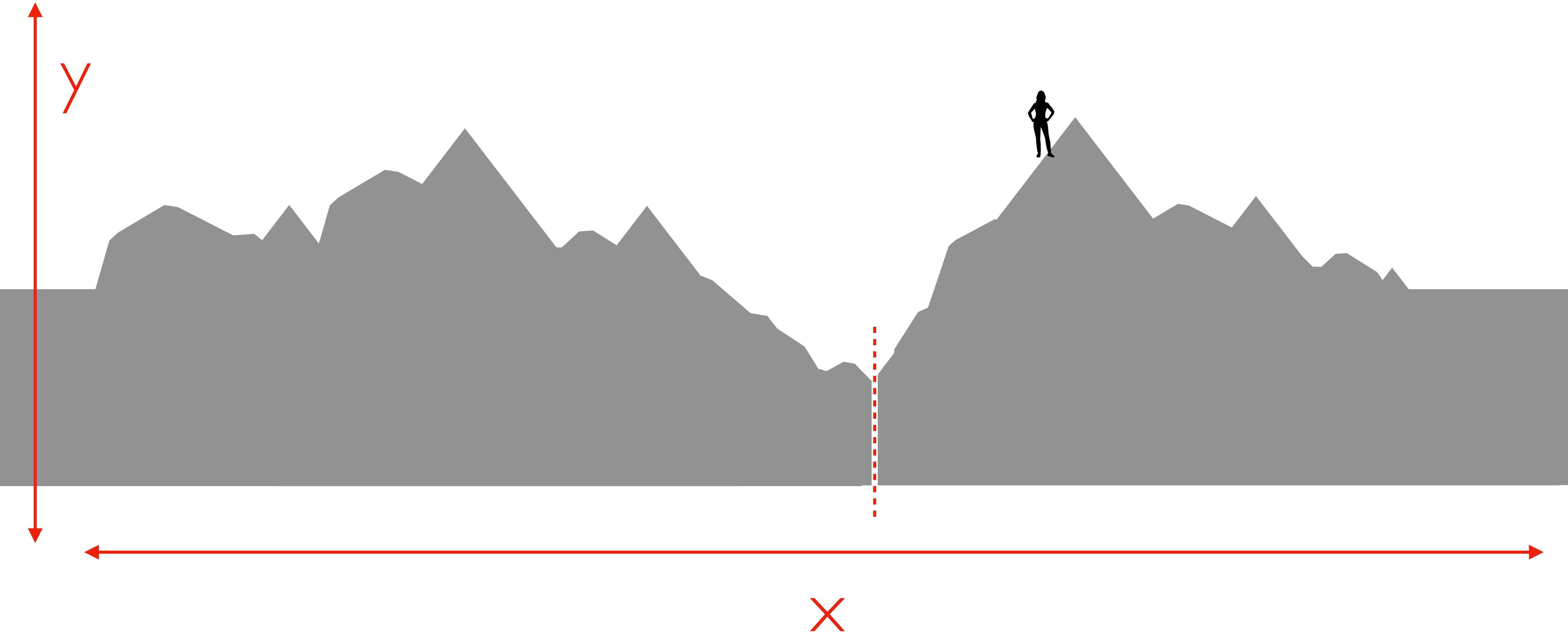
1 find the x value that **minimizes** the y, when y=f(x)



imagine you're in the mountains...

Gradient Descent

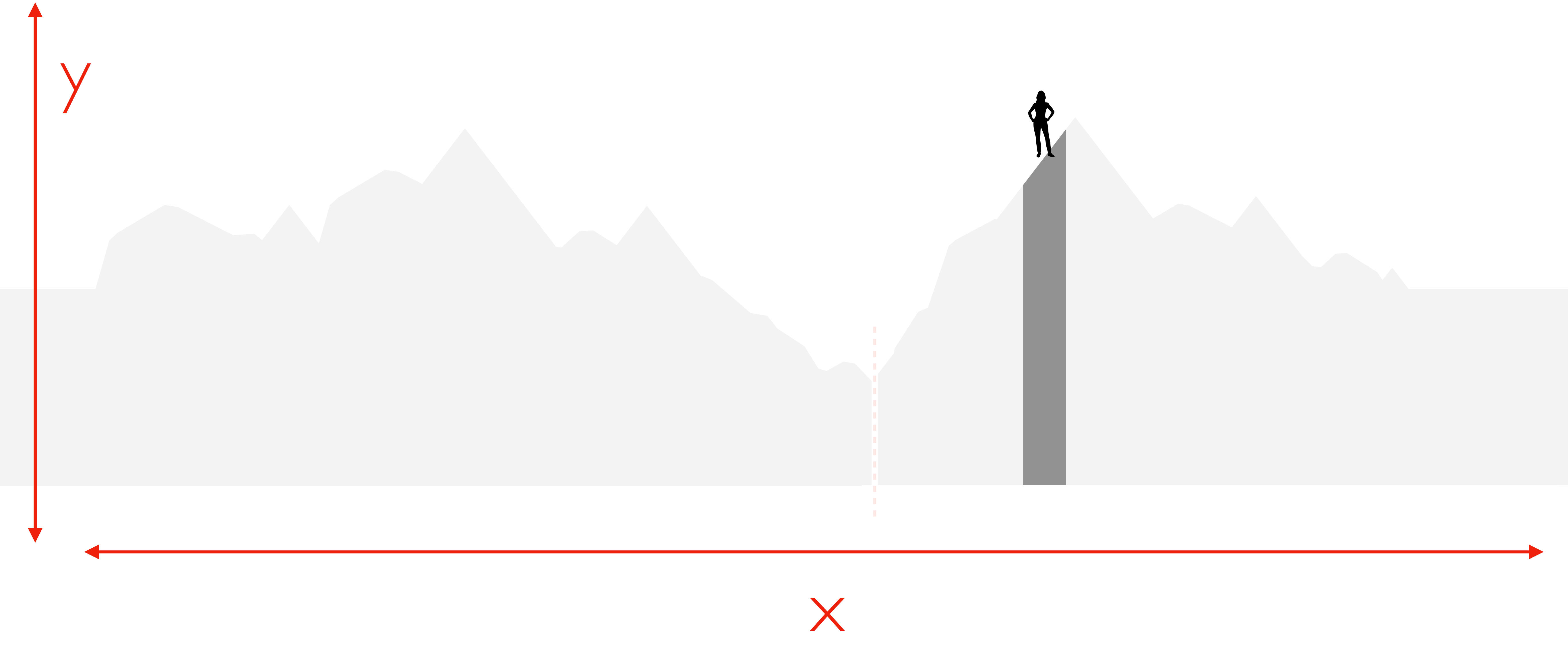
1 find the x value that **minimizes** the y, when y=f(x)



...trying to find the lowest point...

Gradient Descent

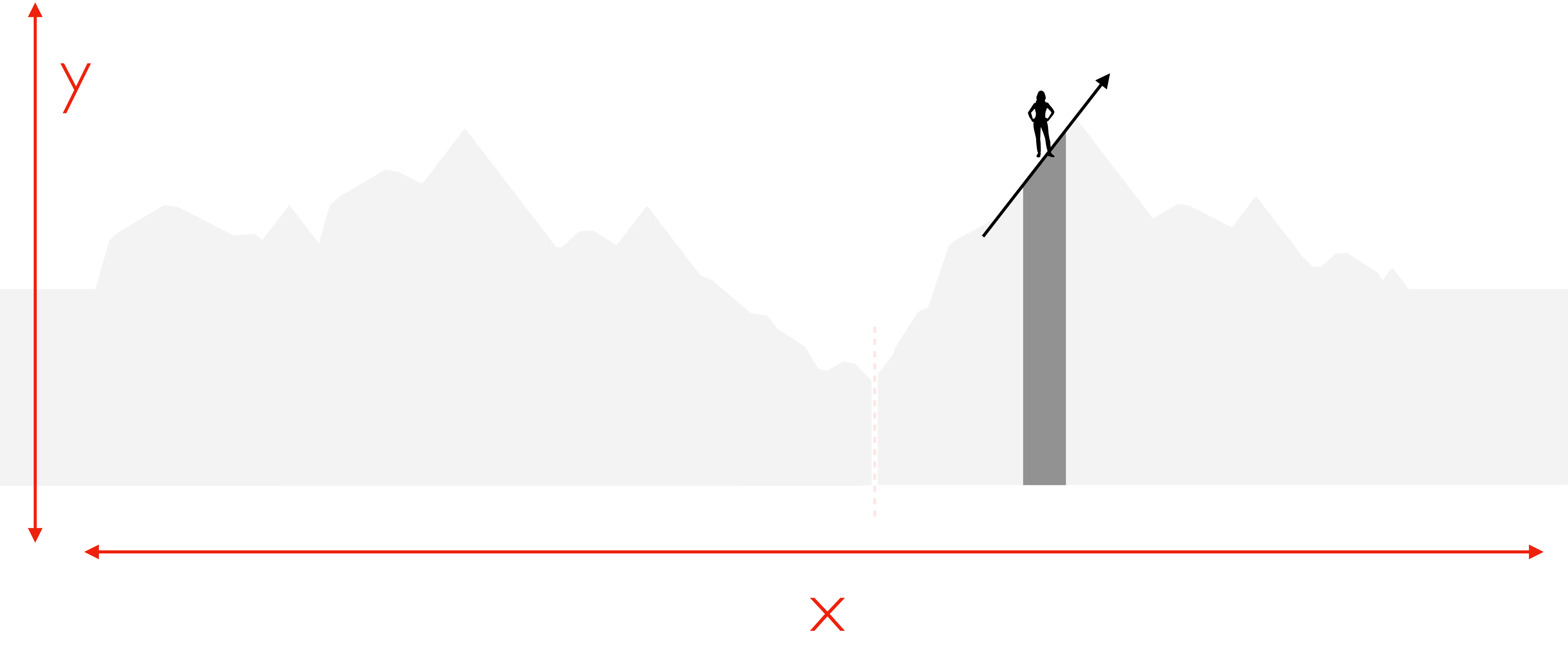
1 find the x value that **minimizes** the y, when y=f(x)



...in a heavy fog

Gradient Descent

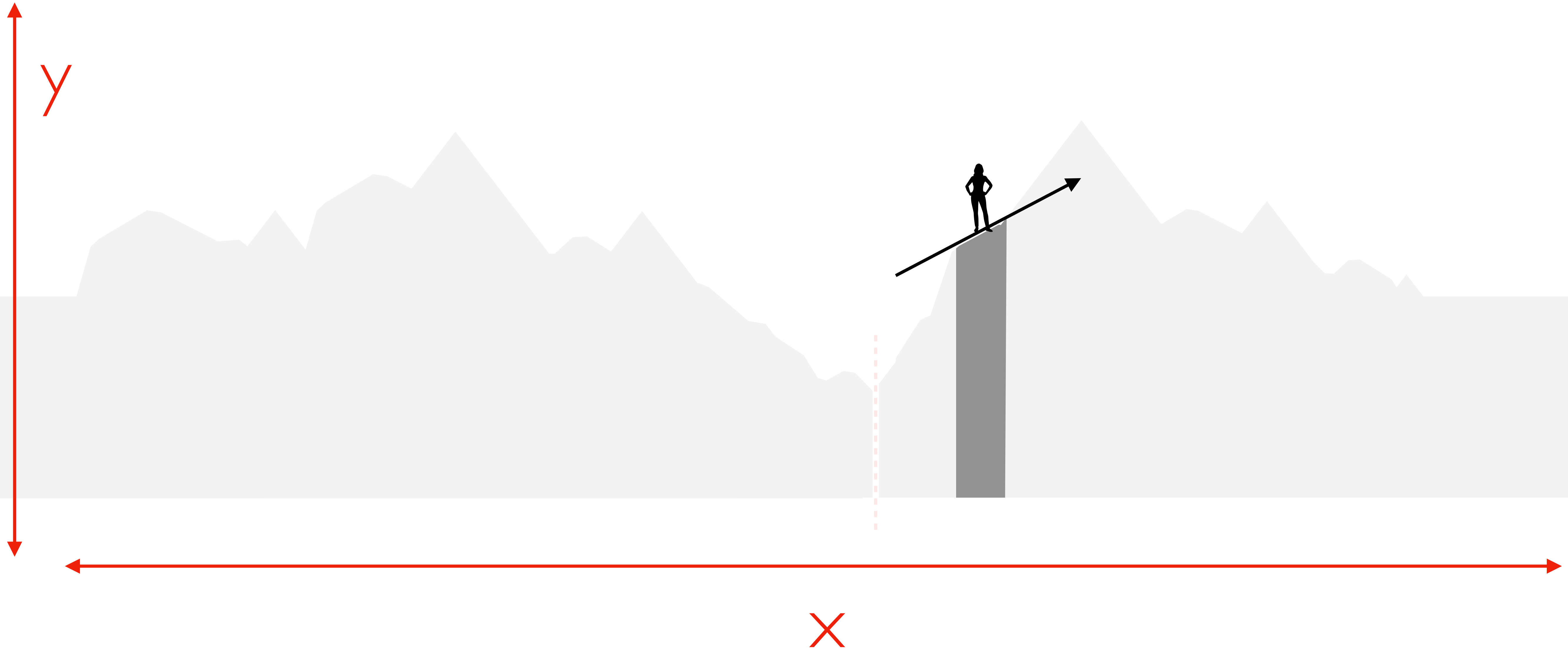
1 find the x value that **minimizes** the y, when y=f(x)



Move to bigger or smaller x? Smaller because the gradient is positive!

Gradient Descent

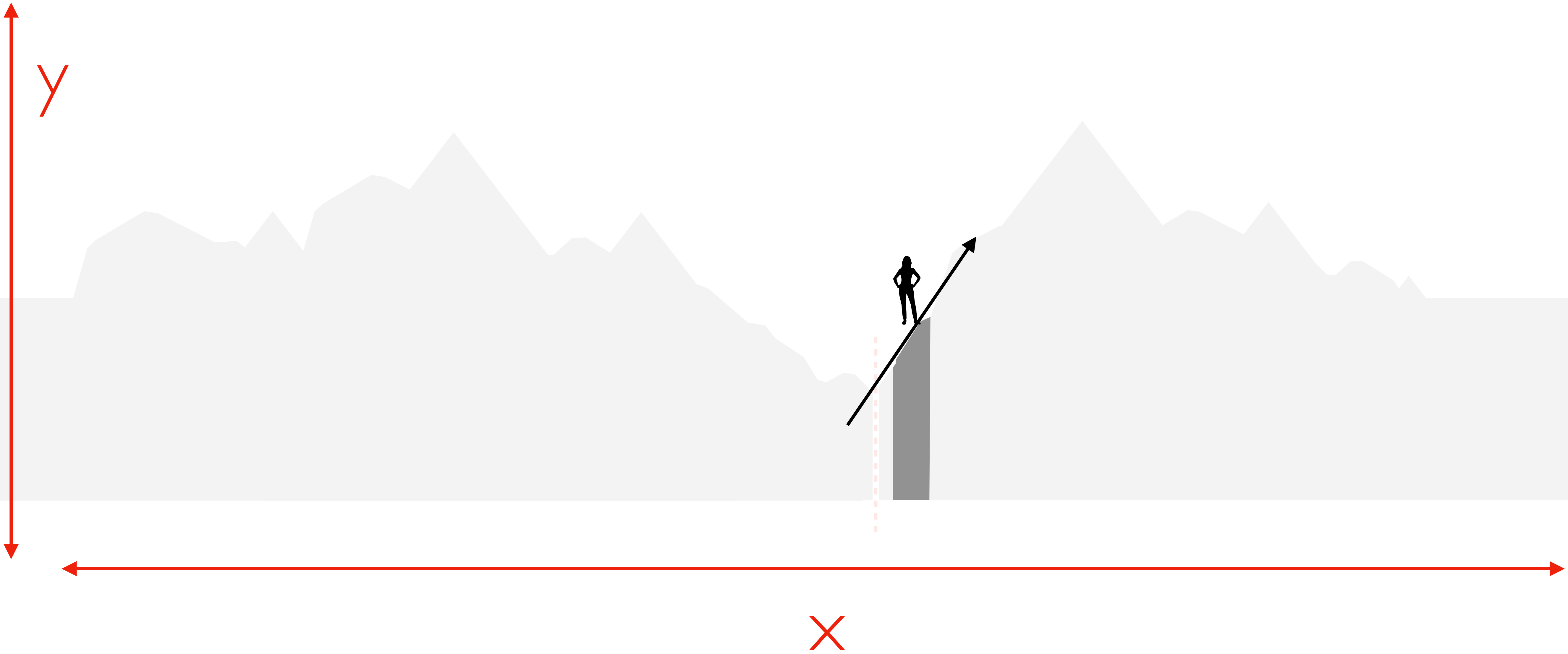
1 find the x value that **minimizes** the y, when y=f(x)



Move to bigger or smaller x? Smaller because the gradient is positive!

Gradient Descent

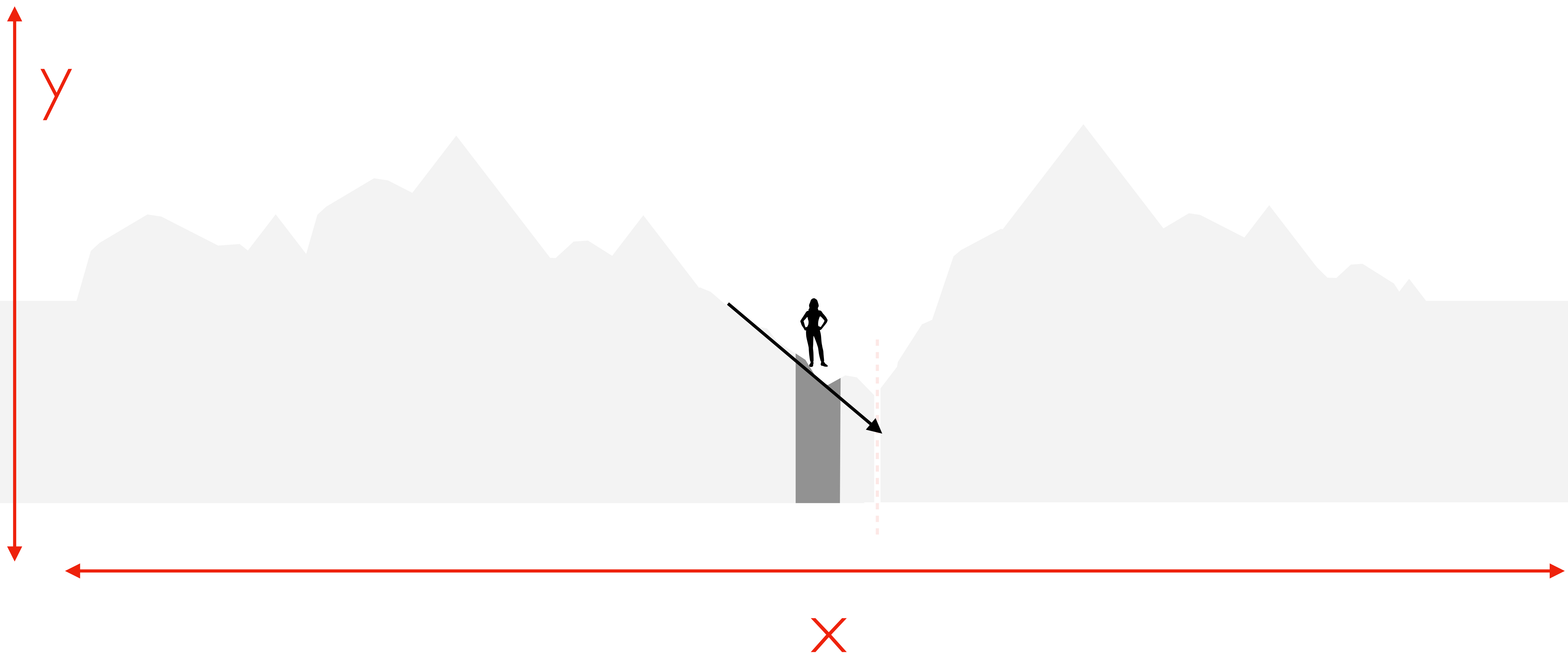
1 find the x value that **minimizes** the y, when y=f(x)



Move to bigger or smaller x? Smaller because the gradient is positive!

Gradient Descent

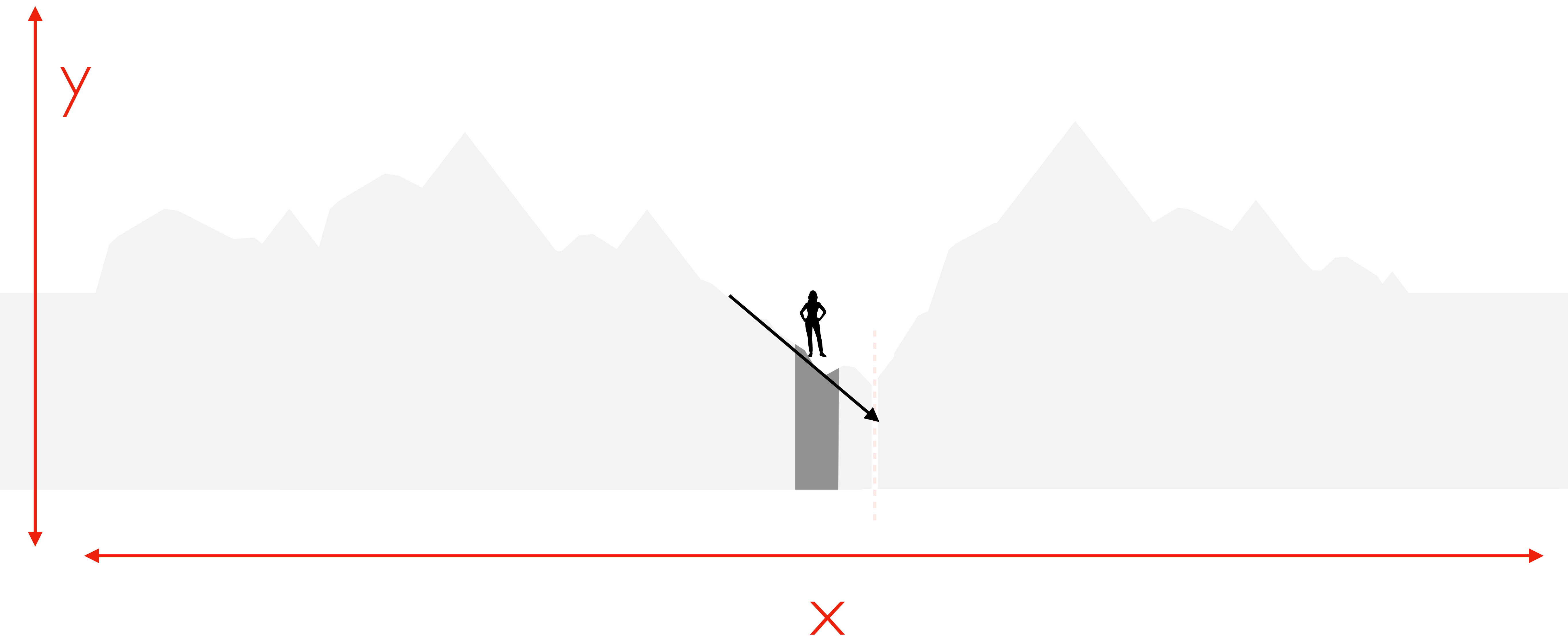
1 find the x value that **minimizes** the y, when y=f(x)



Hiking Analogy Breaks Down: you "Jump" witout crossing area between

Gradient Descent

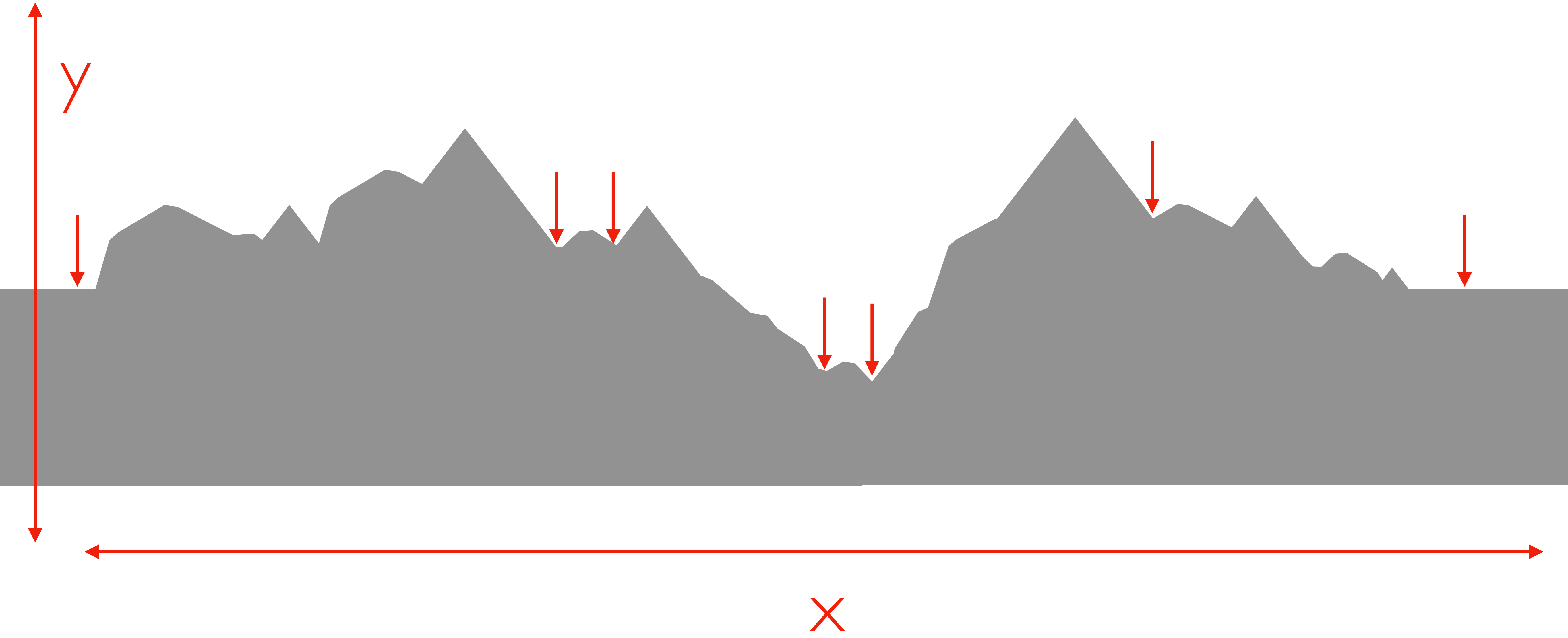
1 find the x value that **minimizes** the y, when y=f(x)



Problem 1: jumpying past the optimimum without realizing it (how far should we jump each time?)

Gradient Descent

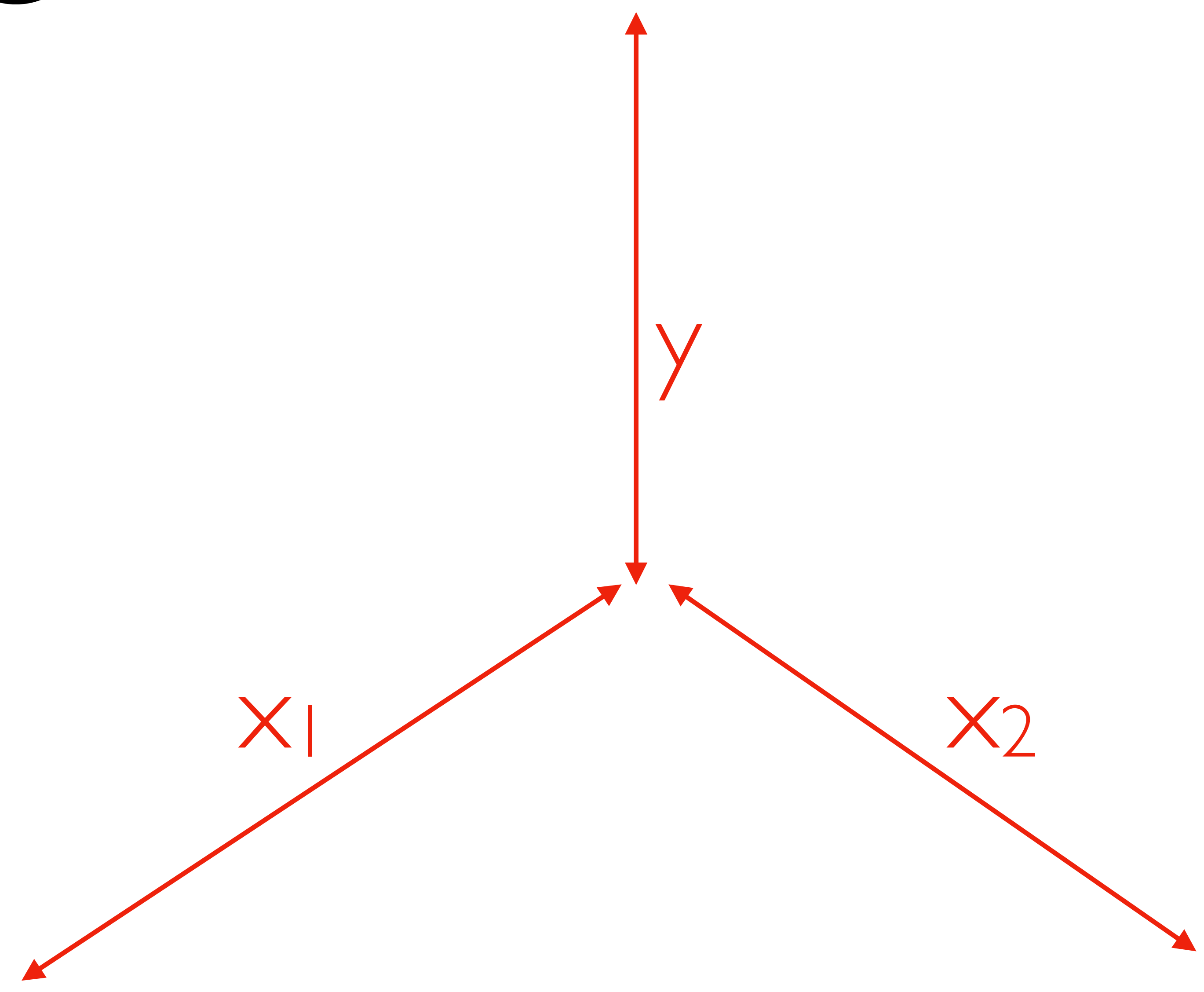
1 find the x value that **minimizes** the y, when y=f(x)



Problem 2: lots of local minima (for certain problems)

Gradient Descent

1 find the x value(s) that **minimize** the y, when $y=f(x_1, x_2)$



Hiking Analogy Breaks Down: there may be MANY dimensions

Gradient Descent

1 find the x value that **minimizes** the y, when $y=f(x_1, x_2, x_3, x_4, ..., x_N)$

?

Hiking Analogy Breaks Down: there may be MANY dimensions

Least Squares, with Gradient Descent

1. find the x value that **minimizes** the y, when $y=f(x_1, x_2, x_3, x_4, ..., x_N)$

2. find the fit line coeficients (slope and intercept) that **minimize** the average squared differences between the data and the line

y = f(x)   where f(x)=slope*x + intercept

error = mean_squared_error(slope, intercept)

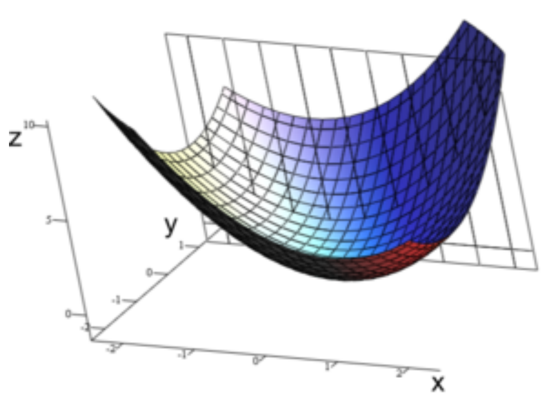
use gradient descent to find best slope, intercept!

Least Squares, with Gradient Descent

1 find the x value that **minimizes** the y, when $y=f(x_1, x_2, x_3, x_4, ..., x_N)$

2 find the fit line coeficients (slope and intercept) that **minimize** the average squared differences between the data and the line

y = f(x) where f(x)=slope*x + intercept



error = mean_squared_error(slope, intercept)

use gradient descent to find best slope, intercept!

mean_squared_error is a convex function: https://en.wikipedia.org/wiki/Convex_function