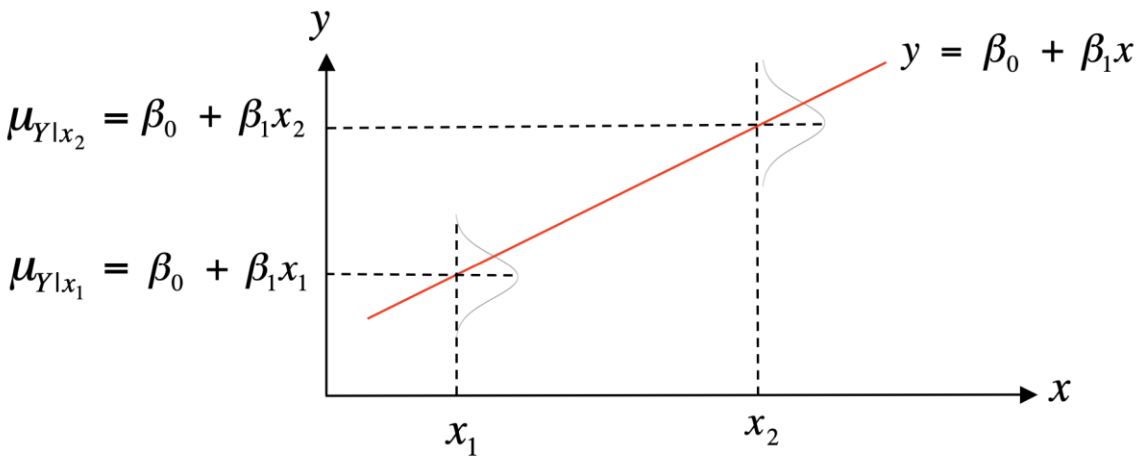
Graphical Interpretation



- For example, if x = height and y = weight then $\mu_{Y|x=60}$ is the average weight for all individuals 60 inches tall in the population

Example

Suppose the relationship between the independent variable height (x) and dependent variable weight (y) is described by a simple linear regression model with true regression line

y = 7.5 + 0.5x and σ = 3

- Q1: What is the interpretation of β1 = 0.5?

The expected change in height associated with a 1-unit increase

in weight.

- Q2: If x = 20 what is the expected value of Y?

$$\mu_{Y|x=20} = 7.5 + 0.5(20) = 17.5$$

- Q3: If x = 20 what is P (Y> 22)?

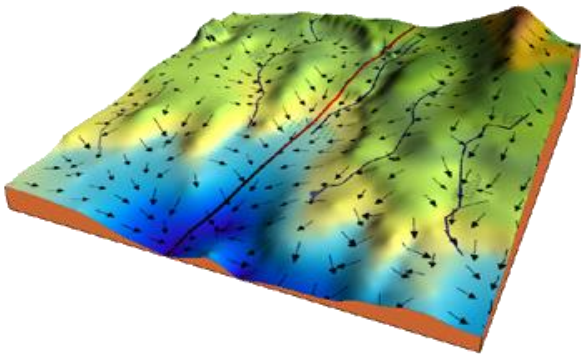
$$P(Y > 22 \mid x = 20) = P\left(\frac{22 - 17.5}{3}\right) = 1 - \phi(1.5) = 0.067$$

Gradient Descent

Gradient descent is an optimization algorithm used to minimize some function by iteratively moving in the direction of steepest descent as defined by the negative of the gradient. In machine learning, we use gradient descent to update the parameters of our model. Parameters refer to coefficients in Linear Regression and weights in neural networks.

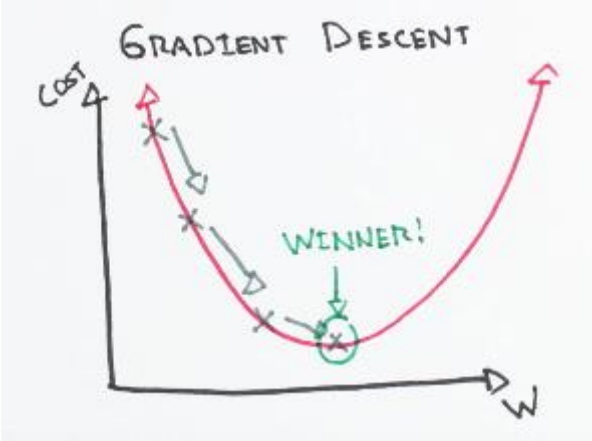
Introduction

Consider the 3-dimensional graph below in the context of a cost function. Our goal is to move from the mountain in the top right corner (high cost) to the dark blue sea in the bottom left (low cost). The arrows represent the direction of steepest descent (negative gradient) from any given point–the direction that decreases the cost function as quickly as possible.



Starting at the top of the mountain, we take our first step downhill in the direction specified by the negative gradient. Next we recalculate the negative gradient

(passing in the coordinates of our new point) and take another step in the direction it specifies. We continue this process iteratively until we get to the bottom of our graph, or to a point where we can no longer move downhill–a local minimum.



Learning rate

The size of these steps is called the *learning rate*. With a high learning rate we can cover more ground each step, but we risk overshooting the lowest point since the slope of the hill is constantly changing. With a very low learning rate, we can confidently move in the direction of the negative gradient since we are recalculating it so frequently. A low learning rate is more precise, but calculating the gradient is time-consuming, so it will take us a very long time to get to the bottom.

Cost function

A Loss Functions tells us “how good” our model is at making predictions for a given set of parameters. The cost function has its own curve and its own gradients. The slope of this curve tells us how to update our parameters to make the model more accurate.