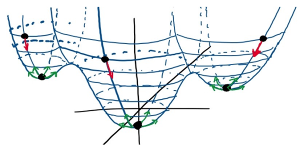
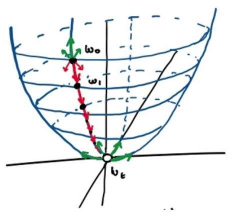
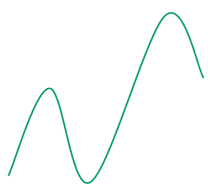
Gradient Descent:

- Intuition:
  - There is no closed form solution to finding the extrema of this cost function. We can however use an iterative process by which we increment w and b gradually toward some minimum (most likely local).
  - **Goal**: find a sequence of w_i's (and b's) that converge toward a minimum.
- Consider a random weight w_0. What happens to Cost(w_0) as you nudge w_0 slightly?
  - As such we can define the following sequence:
    - w_2 = best nudge to w_1
    - w_1 = best nudge to w_0
    - Until we reach w_t that looks like this:
    - 
      - 
    - How can we know how much to nudge and in what direction?
    -
  - Gradients: How can we know how much to nudge and in what direction?
    - $$\nabla f(x) = f'(x)$$
  - 
    - Intuitively, the rate of change of a multi-dimensional function should be a combination of the rate change in each dimension. For a 3-dimensional function, the rate of change would be:
      - $$\nabla f(x, y, z) = \frac{\partial f}{\partial x}\vec{i} + \frac{\partial f}{\partial y}\vec{j} + \frac{\partial f}{\partial z}\vec{k}$$

Example:

$f(x) = 3x^2 - 2y$

Without even computing derivatives we can see that changes in x create more positive change in f than changes in y.

$\nabla f = 6xi - 2j$

This is the gradient of f and can be evaluated at any point (x, y) in the space.

-

$f(x) = 3x^2 - 2y \quad , \quad \nabla f = 6xi - 2j$

Evaluating ∇f at p=(0, 0):

$$\nabla f_p = 6 \cdot 0 \cdot i - 2j = -2j$$

What happens to f as we move 1 unit away from p in the direction of the gradient?

$$p_{new} = 1 \cdot \nabla f_p + p = (0, -2)$$

$$f(p_{new}) = 3 \cdot 0^2 - 2 \cdot (-2) = 4 > f(p) = 0$$

$f(x) = 3x^2 - 2y \quad , \quad \nabla f = 6xi - 2j$

What happens to f if we move 1 unit away from p in a random direction (not following ∇f)? Say (1,0) = 1i + 0j:

$$p_{new} = 1 \cdot (1,0) + p = (1, 0)$$

$$f(p_{new}) = 3 < 4$$

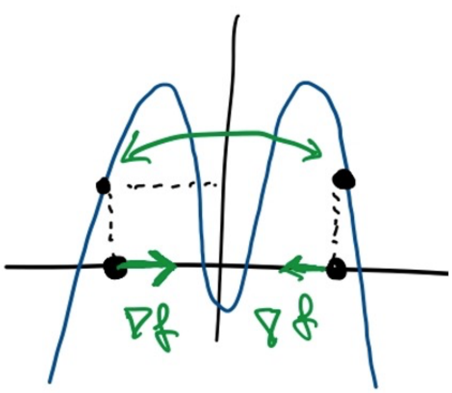
Moving p along the gradient will result in the fastest increase in f from p.

However, the gradient expresses the **instantaneous** rate of change. At p, ∇f_p is the steepest but the highest value of f will depend on how many units we step in that direction. If we step too many units away, the instantaneous change in f is no longer representative of what values f will take.

Example:

However, the gradient expresses the **instantaneous** rate of change. At p, ∇f_p is the steepest but the highest value of f will depend on how many units we step in that direction. If we step too many units away, the instantaneous change in f is no longer representative of what values f will take.

Example:



Given a "smooth" function f for which there exists no closed form solution for finding its **maximum**, we can find a local maximum through the following steps:

1. Define a step size $\boldsymbol{\alpha}$ (tuning parameter)
2. Initialize p to be random
3. $p_{new} = \alpha \nabla f_p + p$
4. p p_{new}
5. Repeat 3 & 4 until $p \sim p_{new}$

To find a local **minimum**, just use $-\nabla f_p$

Notes about **α**:

- If **α** is too large, GD may overshoot the maximum, take a long time to or never be able to converge
- If **α** is too small, GD may take too long to converge

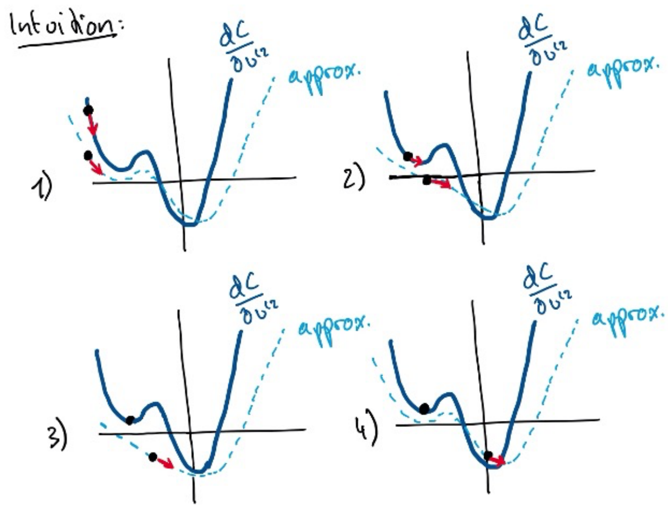
-
- Let's apply this to our diagonal problem to find the weights and bias for logistic regression.
- Assume we have the following dataset:
    - $= \min \operatorname{Cost}(w, b)$
    - $= \min -\frac{1}{n} \sum_{i=1}^{n} \left[yi \log(\sigma(-w^T x_i + b)) + (1 - y_i) \log(1 - \sigma(-w^T x_i + b)) \right]$

We need to compute ∇Cost(w, b):

- $\nabla \operatorname{Cost}(w, b) = \left[\frac{\partial}{\partial w} \operatorname{Cost}, \frac{\partial}{\partial b} \operatorname{Cost} \right]$
- $\frac{\partial}{\partial w} \operatorname{Cost} = \frac{1}{n} \sum_{i=1}^{n} x_i (y_i - \sigma(-w^T x_i + b))$
- $\frac{\partial}{\partial b} \operatorname{Cost} = \frac{1}{n} \sum_{i=1}^{n} \sigma(-w^T x_i + b) - y_i$
- $\frac{\partial}{\partial b} \operatorname{Cost} = \frac{1}{n} \sum_{i=1}^{n} \sigma(-w^T x_i + b) - y_i$

Stochastic Gradient Descent:

- Recall the Cost is computed for the entire dataset. This has some limitations:
- 1. It's expensive to run
- 2. The result we get depends only on the initial starting point

- 

The magnitude of ∇f_p depends on p. A p gets closer to the min / max, the size of ∇f_p decreases.

This also means that points p that contain more “information” have larger gradients. So the order with which this process is exposed to examples matters.