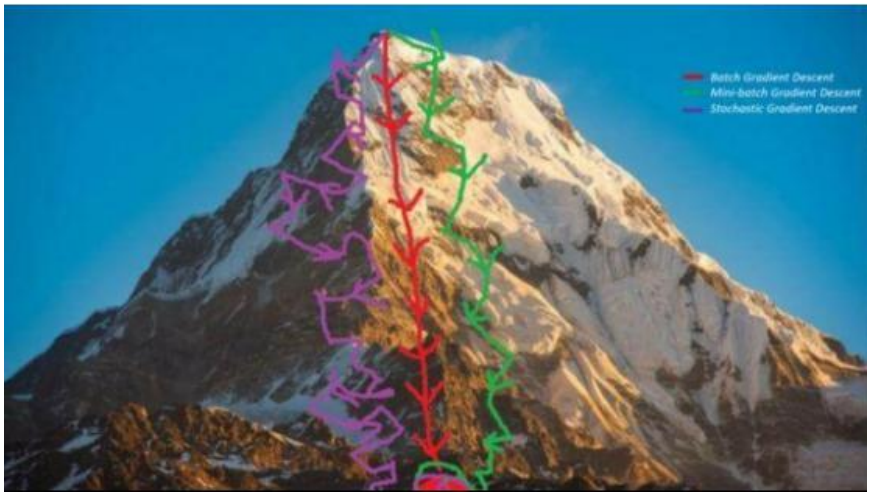
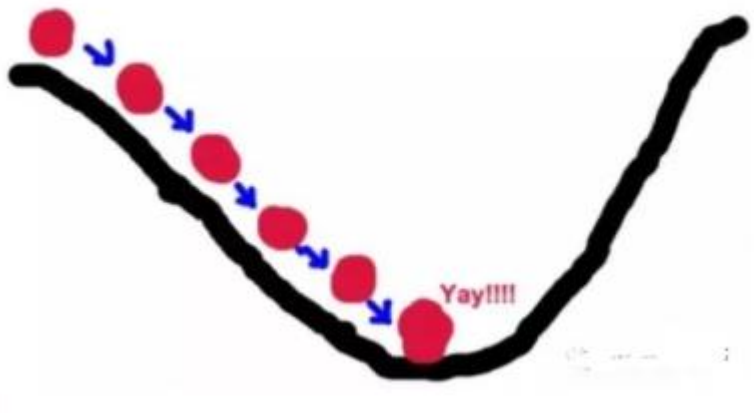
2nd-week-1

Gradient Descent

Wang Jianfang
王建芳(in Chinese)

- Have you ever climbed a mountain? What's the fastest way down the mountain?





Agenda

- Gradient Descent - brief GD
- Example
- Exercise
- Three types of GD
- Homework

What's the GD

- Gradient descent(GD) method, also known as the fastest descent method.
- It was given by the famous mathematician Cauchy in 1847.

What's the GD

- The basic idea
- Suppose we climb a mountain. If we want to get to th e top of the mountain fastest, we should go up the ste epest part of the mountain.That's where the mountain changes the most.
- Similarly, if you start at any point and you need to sear ch the fastest to get the maximum value of the functio n, then you should also search in the direction where t he function changes the fastest.

What's the GD

- What's the fastest way to change the functi on?
- The gradient of the function.

The GD

- If the function is unary, the gradient is the derivative of the function.

$$\nabla f(x) = f'(x)$$

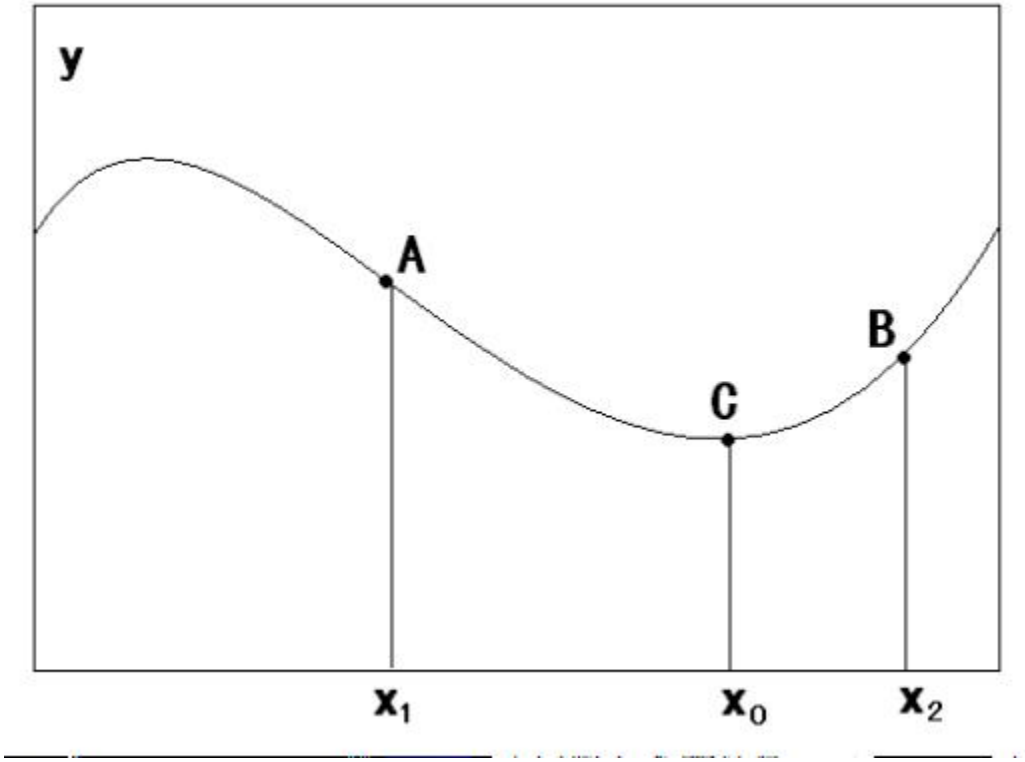
- If it is a binary function, the gradient is defi ned as:

$$\nabla f(x_1, x_2) = \frac{\partial y}{\partial x_1} i + \frac{\partial y}{\partial x_2} j$$

The GD

- If the minimum point of the function is to be found, th en the direction of the negative gradient should be fo und. This method is called **gradient descent** method.
- To search for the minimum point C, the search directio n must be increased to x at point A, which is opposite to the gradient direction at point A.Search must be co nducted in the direction of x decrease at point B, whic h is opposite to the gradient at point B.In short, to sea rch for a minimum, you must search in the direction of the negative gradient.

The GD



Gradient descent method

- Suppose the function $y=f(x_1,x_2...X_n)$ has only one minimum.
- The initial parameter given is $X_0=(x_{10},x_{20}..., x_{n0})$.
- How do I search from this point to find the minim um point of the function?

Methods

- S1: Set a smaller random positive value η, ε。
- S2:Obtain each partial derivative at the current position:

$$f'(x_{m0}) = \frac{\partial y}{\partial x_m}(x_{m0}), m = 1 \sim n$$

- S3: To modify the parameter value of the current function, the formula is as follows:

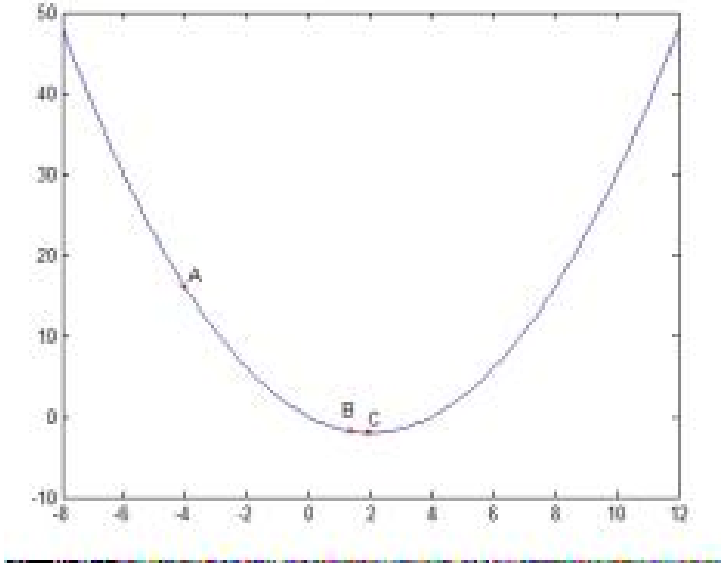
$$x_m^{'} = x_m - \eta \cdot \frac{\partial y}{\partial x_m}(x_{m0}), m = 1 \sim n$$

- S4. If the parameter change is less than ε , exit;
- Otherwise, return S2.

- ***η is Learning rate, ε is accuracy.***

Example1

- For any initial starting point, let's set x0=-4 and use the gradient descent method to fin d the minimus of the function $y=x^2/2-2x$.



Solution

- (1) First, two parameters are given

$$\eta = 0.9, \varepsilon = 0.01$$

- (2) Calculate the derivative

$$\frac{dy}{dx} = x - 2$$

- (3) Calculate the current derivative value

$$y' = -6$$

- (4) Modify the current parameters

$$x' = x - \eta \frac{dy}{dx} = -4 - 0.9 * (-6) = 1.4$$

$$\Delta x = -0.9 * (-6) = 5.4$$

Solution

- (5) Calculate the current derivative value

$$y' = -0.6$$

- (6) Modify the current parameters

$$x' = x - \eta \frac{dy}{dx} = 1.4 - 0.9 * (-0.6) = 1.94$$
$$\Delta x = -0.9 * (-0.6) = 0.54$$

- (7) Calculate the current parameters

$$y' = -0.06$$

- (8) Modify the current parameters

$$x' = x - \eta \frac{dy}{dx} = 1.94 - 0.9 * (-0.06) = 1.994$$
$$\Delta x = -0.9 * (-0.06) = 0.054$$

Solution

- (9) Calculate the current derivative value

$$y' = -0.006$$

- (10) Modify the current parameters

$$x' = x - \eta \frac{dy}{dx} = 1.994 - 0.9 * (-0.006) = 1.9994$$

$$\Delta x = -0.9 * (-0.006) = 0.0054 < \varepsilon$$

- (11)At this point, the change satisfies the terminat ion condition, and the algorithm ends.

Exercise1

- For any initial starting point, let's set x0=-4 and use the gradient descent method to fin d the minimus of the function $y=x^2$.

$$\eta = 0.9, \varepsilon = 0.01$$

Solution

- (1) First, two parameters are given
- (2) Calculate the derivative
- (3) Calculate the current derivative value
-
- (4) Modify the current parameters
-
-

Solution

- (5) Calculate the current derivative value
-
- (6) Modify the current parameters
-
- (7) Calculate the current parameters
- (8) Modify the current parameters

Solution

- (9) Calculate the current derivative value
- (10) Modify the current parameters
- (11)At this point, the change satisfies the terminat ion condition, and the algorithm ends.

3 Types of GD Algorithms

- Batch gradient descent
- Stochastic gradient descent
- Mini batch gradient descent

Batch gradient descent

- The Batch gradient Descent is an iteration to train all the samples, which is an endless iteration. The solution idea of the whole algorithm can be expres sed as:
- (1) Take the partial derivative of J(Theta) with respect to theta, and get the gradient corresponding to each theta:

$$\frac{\partial J(\theta)}{\partial \theta_j} = -\frac{1}{m}\sum_{i=1}^{m}(y^i - h_\theta(x^i))x_j^i$$

Batch gradient descent

- (2) To minimize the risk function, update each Theta according to the negat ive direction of the gradient of each parameter, Theta:

$$\theta_j' = \theta_j + \frac{1}{m}\sum_{i=1}^{m}(y^i - h_\theta(x^i))x_j^i$$

- (3) It can be noted from the above formula that it can obtain a global optim al solution, but every iteration step needs to use all the data of the training s et. If M is large, then you can imagine the iteration speed of this method!! So, this introduces another method, stochastic gradient descent.

Stochastic gradient descent

- To accelerate the convergence speed and solve the pro blem that large amounts of data cannot be inserted in to memory at one time, the stochastic Gradient Desce nt (SGD) is put forward. The idea of SGD is to train onl y one sample at a time to update the parameters. Spe cific implementation ideas:

Stochastic gradient descent

- (1) The above risk function can be written in the following form. The loss function corresponds to the granularity of each sample in the training set, while the above batch gradient descent corresponds to all the training samples:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}(y^i - h_\theta(x^i))^2 = \frac{1}{m}\sum_{i=1}^{m}\cos t(\theta,(x^i,y^i))$$

$$\cos t(\theta,(x^i,y^i)) = \frac{1}{2}(y^i - h_\theta(x^i))^2$$

Stochastic gradient descent

- 2) The loss function of each sample, obtain the corresp onding gradient by taking the partial derivative of Thet a, to update Theta:

$$\theta_j^{'} = \theta_j + (y^i - h_\theta(x^i))x_j^i$$

Stochastic gradient descent

- To accelerate the convergence speed and solve the problem that large amounts of data cannot be inserted into memory at one ti me, the stochastic Gradient Descent (SGD) is put forward. The idea of SGD is to train only one sample at a time to update:

Randomly shuffle (reorder) training examples

Repeat {
  for $i := 1, \ldots, m$ {
    $\theta_j := \theta_j - \alpha(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$
    (for every $j = 0, \ldots, n$)
  }
}

Mini batch gradient descent

- Mini-batch gradient Descent is the compromise plan for the Ba tch gradient and stochastic Gradient Descent, that is, a part of t he samples are used to update the parameters each time.

Repeat{

for i=1, 11, 21, 31, ... , 991{

$$\theta_j := \theta_j - \alpha \frac{1}{10} \sum_{k=i}^{i+9} (h_\theta(x^{(k)}) - y^{(k)}) x_j^{(k)}$$

(for every j=0, ... , n)

}

}

Mini batch gradient descent

- Let's say we use 10 samples per parameter update (different tas ks are completely different, just to give you an example)
- Update the pseudocode as follows:

Repeat{

for i=1, 11, 21, 31, ... , 991{

$$\theta_j := \theta_j - \alpha \frac{1}{10} \sum_{k=i}^{i+9} (h_\theta(x^{(k)}) - y^{(k)}) x_j^{(k)}$$

(for every j=0, ... , n)

}

}

Homework

- Giving the details of the process using the gradient descent meth od int the notebook.
- y = 0.5 * (x-0.25) ** 2
- inital value:
- η=0.500
- ε=1e-10

Questions and Comments?

Thank you!!