Week 4 Lab

Bookmark this page

Week 4 Lab

(1.0 points possible)

Gradient descent

This week's lab and homework explore the concept of machine learning as optimization, building on the lecture and lecture notes on margin maximization and gradient descent.

Group information

1) Explore gradient descent

We've established that machine learning problems can be posed as optimization problems. We begin by studying general strategies for finding the minimum of a function. In general, unless the function is convex, it may be computationally difficult to find its global minimum.

Note: A function is convex if the line segment between any two points on the graph of the function lies above or on the graph.

We will sometimes study convex objectives, but in other cases we will content ourselves with finding a local minimum (where the gradient is zero) which may not be a global minimum. One method to find a local minimum of a function is called gradient descent (there are better ways, but this one is simple and computationally efficient in high dimensions and with lots of data). The idea is that we start with an initial guess, x_0, and move "downhill" in the direction of the gradient, leading to an update step

$$x^{(1)} = x^{(0)} - \eta \nabla_x f(x^{(0)})$$

where η is a "step size" parameter with the constraint that $\eta > 0$. We continue updating until $x^{(i+1)}$ does not differ too much from $x^{(i)}$. This approach is guaranteed to find the minimum if the function is convex and η is sufficiently small.

The questions below are concerned with running gradient descent on the parabola

$$f(x) = (2x+3)^2 \, .$$

1A) Formulate the update rule that will be executed on every step when performing gradient descent on $f(x)$. You may use `eta` and `x` in your Python expression, where `eta` represents η.

x ← x - eta * 4 * (2 * x + 3)

Check Syntax | Submit | View Answer | Ask for Help 100.00%

You have infinitely many submissions remaining.

Your entry was parsed as:

$$x - \eta \times 4 \times (2 \times x + 3)$$

1B)

What is the optimal value of x that minimizes f? -1.5

Submit | View Answer | Ask for Help 100.00%

You have infinitely many submissions remaining.

1C) This question asks you to explore convergence of gradient descent for our $f(x)$, to see how the step size affects the rate of convergence and whether there are oscillations or lack of convergence.

We implement minimization of $f(x)$ using a function `t1`, which runs gradient descent for the minimization of $f(x)$. We halt the algorithm when the value of x changes by less than 10^{-5}. In general, we may use some small tolerance such as this to say whether gradient descent has *converged*. Conversely, *divergence* is defined as being when x will not converge to a single finite value, even with an infinite number of updates.

When you click Submit, the tutor question generates a plot of $f(x)$ in blue and the history of x values you have tried in red with a blue 'x' at the initial x value. You can change the values for `step_size` or `init_val` and click Submit again (all submits get 100%).

Experiment with the following step sizes: [0.01, 0.1, 0.2, 0.3]. For which one(s) does $x^{(k)}$ converge without oscillation? For which one(s) does $x^{(k)}$ diverge?

```
def run():
    return t1(step_size= 0.125, init_val = 0)

```

Run Code | Submit | View Answer | Ask for Help 100.00%

You have infinitely many submissions remaining.

Your score on your most recent submission was: 100.00%

Test Results:

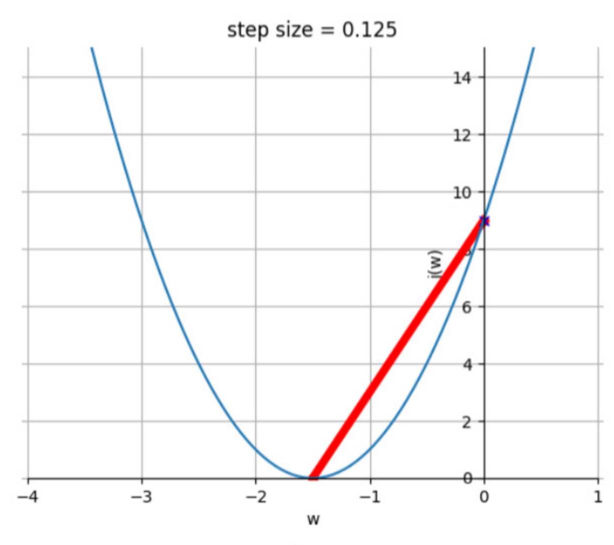
Test 01

The test case was:

```
#Your Code Here
ans=run()
```

Our solution produced the following value for `ans`:

Your submission produced the following value for `ans`:



('objective', 0.0, 'weights', array([[-1.5]]))

Why do oscillations happen for some values of `step_size` and not others? To gain insight into this behavior (and to understand how gradient descent proceeds), we note that we can rewrite our update rule:

$$x^{(k+1)} = x^{(k)} - \eta \nabla_x f(x^{(k)})$$

in the form:

$$z^{(k+1)} = \alpha z^{(k)}$$

for some $z = x - C$, and then consider how or if z approaches 0 for large k (or equivalently, x approaches C for large k). In particular, this formulation will soon be useful for us to analyze how the step-size η affects convergence.

1D) Show that the gradient descent update rule for our function $f(x) = (2x+3)^2$ can be written in the form:

$$x^{(k+1)} + 3/2 = (1 - 8\eta)(x^{(k)} + 3/2)$$

and equivalently in the form

$$z^{(k+1)} = \alpha z^{(k)}$$

by defining $z^{(k)}$ and α appropriately. What is $z^{(k)}$ in terms of $x^{(k)}$? What is α in terms of η?

Written in this form, we make the following key observation: **we can think of our gradient descent step as simply a multiplication of the previous value by α on each step.**

Inductively, we can see that the value of z (and x) at step k is related to the initial value as

$$z^{(k)} = \alpha^k z^{(0)}$$

and equivalently

$$x^{(k)} + 3/2 = (1 - 8\eta)^k (x^{(0)} + 3/2) \, .$$

(Note: if this is not clear, please feel free to join the help queue.)

The above equations are useful because they explain the relationship between the initial value of z (or x) and the output of gradient descent as the repeated multiplication by the same factor α on each step. From this, we can identify cases of convergence, oscillation, and divergence by the following values on α:

$\alpha > 1$	**Gradient descent diverges without oscillation; $z \to \infty$**
$\alpha = 1$	**$z^{(k)} = z^{(0)}$, so no gradient descent steps occur**
$1 > \alpha \geq 0$	**α^∞ approaches 0, so gradient descent converges; $z \to 0$**
$0 > \alpha > -1$	**α^∞ approaches 0 while changing signs every step, so converges with some oscillation**
$\alpha = -1$	**At every step, the sign of z flips. Gradient descent oscillates between $z^{(0)}$ and $-z^{(0)}$ endlessly**
$-1 > \alpha$	**Gradient descent diverges with oscillation, since z grows but the sign of z flips at every step**

Since our ultimate goal is convergence, we are interested in the cases where $|\alpha| < 1$.

We can use the rules above about α to reason about how η affects convergence, given that $\alpha = 1 - 8\eta$.

Answer the following questions algebraically (using the expressions above).

1E)

What is the range of step size, as an interval, that causes x to converge to the global minimum starting from an arbitrary initial value? Use ( ) for open intervals and [ ] for closed intervals

(0, 0.25)

Submit | View Answer | Ask for Help 100.00%

You have infinitely many submissions remaining.

Does your algebraic answer agree with your numerical experiments above?

1F)

What is the largest step size that causes x to converge without oscillating?

0.125

Submit | View Answer | Ask for Help 100.00%

You have infinitely many submissions remaining.

How many iterations are needed for convergence in this case? Run `t1` with this value of step size.

1G)

For a step size of 0.1, is there an initial value of x_0 for which x does not converge to the global minimum?

No

Ask for Help 100.00%

You have infinitely many submissions remaining.

Solution: No

Explanation:

As discussed above, x converges for all step sizes in the range $(0, 0.25)$. The point of convergence is a local minimum of f, but f is convex and admits a unique minimum.

1H) What value(s) of step size in the set $\{0.1, 0.11, 0.12, 0.13, 0.14, 0.15\}$ makes gradient descent take the most steps before convergence? Find the answer algebraically and enter your value(s) in a Python list. (Hint: think about the magnitude of $1 - 8\eta$ and how this might affect the rate of convergence).

Enter your answer as a python list:

Submit | View Answer | Ask for Help

You have infinitely many submissions remaining.

Now try running `t1` with the above step sizes. Which ones are slowest? Which ones oscillate? Do these behaviors match with your algebraic results?

2) Where to meet?

A group of friends is planning to host a baby shower over the weekend. They want to find a location for the party that minimizes the sum of *squared* distances from their houses to the location of the party. Assume for now that they can host the party at any location in the town.

Assuming that the friends live in a 1-dimensional town, solve the following problems:

2A) Pose this problem as an (unconstrained) optimization problem. Assume there are n friends and the i-th friend is located at l_i. Denote the location of the party by p. What is the objective as a function of p? Write it down.

2B) Compute the gradient (write down/show your computation). Where is it zero?

2C) Which of the following is true?

- [x] There is necessarily a unique location that minimizes the objective function
- [ ] The optimization problem may have local minima that are not global minima
- [ ] The party can be always be hosted in one of the houses without loss of optimality
- [x] There is necessarily a choice of step size that makes gradient descent converge with oscillations for this problem

Ask for Help 100.00%

You have infinitely many submissions remaining.

Solution:

✔ There is necessarily a unique location that minimizes the objective function

✘ The optimization problem may have local minima that are not global minima

✘ The party can be always be hosted in one of the houses without loss of optimality

✔ There is necessarily a choice of step size that makes gradient descent converge with oscillations for this problem

Explanation:

Discuss with your partner, and be ready to explain your answers and reasoning!