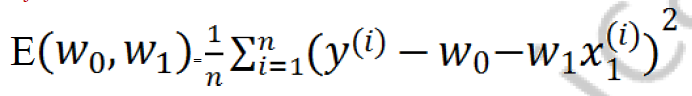
**Practical Data Science (Optimization)**

**Problem 1: Applying gradient descent variations for ML objective function**

As we discussed in class, machine learning objective functions are defined based on error or loss induced by train data while learning. Assume the following objective function and train data for learning.

Objective function:



Train data:

|  |  |  |
| --- | --- | --- |
| S.n | x1 | y |
| 1 | 10 | 40 |
| 2 | 30 | 60 |
| 3 | 40 | 70 |
| 4 | 50 | 90 |
| 5 | 60 | 80 |
| 6 | 70 | 70 |

Do the following:

a) Compute the gradient of E.

**[Ans:]** The gradient of E is the vector of partial derivatives of E with respect to w0 and w1.

E(w0,w1) = -2(y(i) – w0 – w1x1(i))

E(w0,w1) = -2x1(y(i) – w0 – w1x1(i))

Gradient = [-2(y(i) – w0 – w1x1(i)), -2x1(y(i) – w0 – w1x1(i))]

Applying batch gradient descent Algorithm

b) Write the expressions for weight updates using batch gradient descent algorithm.

**[Ans:]**

Assuming step-factor α = 0. 00007.

w0 = w0 - αE(w0,w1); w1 = w1 - αE(w0,w1)

In batch descent, we iterate over all the ‘n’ inputs to compute the partial derivatives. Given n = 6 inputs. So, we have

w0 = w0 – 0. 00007\*[-2(y(i) – w0 – w1x1(i))]

w1 = w1 – 0. 00007\*[-2 x1(i) (y(i) – w0 – w1x1(i))]

c) Simulate (by hand) the batch gradient descent algorithm for 5 iterations. For each iteration, show the error value and updated weights. Assume initial values of (w0,w1) as (0,0).

**[Ans:]**

*See the logic for the below numbers in the accompanying Assignment8-Optimization.R file.*

The largest α for which there is convergence is about 0.00007. So, for the below calculations, we use α = 0.00007.

Initial: wo = 0.000000, w1 = 0.000000, error = 4916.666667

Iteration 1: wo = 0.057400, w1 = 2.688000, error = 4096.443599

Iteration 2: wo = 0.016909, w1 = 0.255959, error = 3425.096172

Iteration 3: wo = 0.064977, w1 = 2.455998, error = 2875.599701

Iteration 4: wo = 0.032925, w1 = 0.465413, error = 2425.835163

Iteration 5: wo = 0.073356, w1 = 2.266068, error = 2057.698697

.

.

.

Iteration 80000: wo = 41.713874, w1 = 0.614294, error = 100.476191

Sum of residuals = y(i) – w0 – w1x1(i)

Sufficiently accurate values of w0 and w1 (correct to 2 decimal places) are attained after 80000 iterations, where the sum of residuals is 0.0004233926.

For the line of best fit under linear regression, the sum of residuals is always 0. We can thus be confident that our final solution/approach is correct.

Note that the error here will not become 0 as it is the sum of ‘square’ of residuals.

Applying stochastic gradient descent Algorithm

d) Write the expressions for weight updates using stochastic gradient descent algorithm.

**[Ans:]**

w0 = w0 - αE(w0,w1); w1 = w1 - αE(w0,w1)

In stochastic gradient descent, we use just 1 random input in each iteration to compute the partial derivatives. Assuming α = 0. 00007. Using n = 1 since we use only 1 input in each iteration. So, we have

w0 = w0 – 0. 00007\*[-2(y(i) – w0 – w1x1(i))]

w1 = w1 – 0. 00007\*[-2 x1(i) (y(i) – w0 – w1x1(i))]

e) Simulate (by hand) the stochastic gradient descent algorithm for 4 iterations. For each iteration, show the error value and updated weights. Assume initial values of (w0, w1) as (0,0) and random samples picked are 3, 2, 5, 1.

**[Ans:]**

*See the logic for the below numbers in the accompanying Assignment8-Optimization.R file.*

Assuming α = 0. 00007.

Initial: wo = 0.000000, w1 = 0.000000, error = 4916.666667

Iteration 1: wo = 0.009800, w1 = 0.392000, error = 2755.165435

Iteration 2: wo = 0.016552, w1 = 0.594567, error = 1911.318664

Iteration 3: wo = 0.022756, w1 = 0.966766, error = 846.670632

Iteration 4: wo = 0.026999, w1 = 1.009200, error = 765.025018

Applying stochastic mini-batch gradient descent Algorithm

f) Write the expressions for weight updates using stochastic mini-batch gradient descent algorithm.

**[Ans:]**

w0 = w0 - αE(w0,w1); w1 = w1 - αE(w0,w1)

In stochastic mini-batch descent, we use a subset k=batch size of inputs in each iteration to compute the partial derivatives. Assuming α = 0. 00007. Given batch size of k = 3. So, we have

w0 = w0 – 0. 00007\*[-2(y(i) – w0 – w1x1(i))]

w1 = w1 – 0. 00007\*[-2 x1(i) (y(i) – w0 – w1x1(i))]

g) Simulate (by hand) the stochastic mini-batch gradient descent algorithm for 3 iterations. For each iteration, show the error value and updated weights. Assume initial values of (w0,w1) as (0,0) and random batches of size 3 as follows - (1,4,5), (2,3,6) across iterations.

**[Ans:]**

*See the logic for the below numbers in the accompanying Assignment8-Optimization.R file.*

Assuming α = 0. 00007.

Initial: wo = 0.000000, w1 = 0.000000, error = 4916.666667

Iteration 1: wo = 0.029400, w1 = 1.358000, error = 405.014782

Iteration 2: wo = 0.030771, w1 = 1.280536, error = 437.264496

Iteration 3: wo = 0.038645, w1 = 1.526514, error = 428.698257

Note that we should not keep using the same set of batches across iterations – (1,4,5) and (2,3,6) in an alternating fashion. Otherwise, the values will convergence to the sub-optimal value of wo = 42.048718 and w1 = 0.480035, resulting in a sum of residuals value of 32.89858.

**Problem 2: Find the least amount of fencing**

A rectangular paddock is having an area of 50 m2. One side of the rectangle is straight wall as shown below and the remaining three sides are to be made from wire fencing. Do the following:



a. Find the expression for required fencing

**[Ans:]** Let the length = l, breadth = b.

Given that area = l\*b = 50 m2

Perimeter = l + 2b = (50/b) + 2b

b. Find the least amount of fencing required using calculus approach.

**[Ans:]** Need to minimize f(b) = 2b + (50/b).

For minimum f(b), f’(b) = 0 = 2 – 50/b2

Solving this gives b = 5, -5. Since b > 0, we take b = 5.

So, the least amount of fencing needed = f(5) = (2\*5) + (50/5) = 20 m

**Problem 3: Design maximum volume open rectangular box**

The diagram below shows a 24cm by 15cm sheet of cupboard from which a square of side x cm has been removed from each corner. The cardboard is then folded to form an open rectangular box of depth x cm and volume of v cm3. Do the following:



a. Find the expression for volume

**[Ans:]**

Volume = length \* breadth \* depth

Depth = x

Length = 24 – 2x

Breadth = 15 – 2x

So, volume = (24 – 2x)(15-2x)x = 2x(12-x)(15-2x)

= 2x(180 – 24x – 15x + 2x2)

= 2x(180 – 39x + 2x2)

= 4x3 – 78x2 + 360x

b. Find value of x for which volume is maximum using calculus approach

**[Ans:]**

Need to minimize f(x) = 4x3 – 78x2 + 360x

f’(x) = 0 = 12x2 – 156x + 360 = 12 (x2 – 13x + 30) = 12 (x-10)(x-3)

Solving, we get x = 3 or x = 10.

Obviously, x = 10 is not possible because, then, breadth = 15 – 2x < 0

So, **x = 3** is the solution.

**Problem 4: Design of optimal sized petrol tank**

An emergency petrol tank is designed to carry 1 gallon of petrol (4546 cm3). Its shape can be considered to be cuboid as shown below. The base of the cuboid is a rectangle with the length double the width. Do the following:



a. Find the expression for surface area of tank

**[Ans:]**

Given that the cuboid base is a rectangle with length double the width.

Let width = w. So, length = 2w. Let height = h.

Volume = 4546 = width\*length\*height = w\*2w\*h

Surface area = 2(2wh + wh + w\*2w) = 6wh + 4w2

= 4w2 + 6w(4546/2w2)

= 4w2 + (13638/w)

b. Find the dimensions of tank that minimizes the required surface area using calculus approach

**[Ans:]**

Need to minimize f(w) = 4w2 + (13638/w)

f’(w) = 0 = 8w – 13638/w2

So, 8w = 13638/w2

w3 = 13638/8 = 1704.75

Width = w = (1704.75)1/3 = 11.946

Length = 2w = 23.892

Height = 4546/2w2 = 2273/w2 = 15.928

So, for minimum surface area, the required dimension (in cm) is as below:

**11.946 x 23.892 x 15.928**