BATCH-STOCHASTIC SUB-GRADIENT METHOD FOR SOLVING NON-SMOOTH CONVEX LOSS FUNCTION PROBLEMS

KasimuJuma Ahmed

Mathematics Unit, Department of General Studies, Federal Polytechnic Bali,
Taraba State, Nigeria.

ABSTRACT

Mean Absolute Error (MAE) and Mean Square Error (MSE) are machine learning loss functions that not only estimates the discrepancy between prediction and true label but also guide the optimal parameter of the model.Gradient is used in estimating MSE model and Sub-gradient in estimating MAE model. Batch and stochastic are two of the many variations of sub-gradient method but the former considers the entire dataset per iteration while the latter considers one data point per iteration. Batch-stochastic Sub-gradient method that learn based on the inputted data and gives stable estimated loss value than that of stochastic and memory efficient than that of batch has been developed by considering defined collection of data-point per iteration. The stability and memory efficiency of the method was tested using structured query language (SQL). The new method shows greater stability, accuracy, convergence, memory efficiencyand computational efficiency than any other existing method of finding optimal feasible parameter(s) of a continuous data.

KEYWORDS

Machine learning, Loss function, sub-gradient,*Mean Absolute Error (MAE) and Prediction*

1. INTRODUCTION

An optimization (i.e., minimize cost of production or maximize price) problem is of the form

$$\underset{\text{b}, w_1, w_2, \dots, w_m}{min} \text{f(x)} = \text{J (b}, w_1, w_2, \dots, w_m). \qquad (1)$$

Equation (1) can be solved using model parameters [2].

One of the tools in estimating model parameter is loss function. The most common used model parameter for estimating loss function is Mean Square Error (MSE)

$$\text{J (b}, w_1, w_2, \dots, w_m) = \frac{1}{2}\sum_{i=1}^{n} \left(h_{b,w_1, \dots, w_m}\left(x^{(i)}\right) - y^{(i)}\right)^2 \qquad (2)$$

which if use, produces a bowl shape curve at the learning stage.

David C. Wyld et al. (Eds): CSTY, AIMLNET, AI, BINLP -2023
pp. 65-84, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.131806

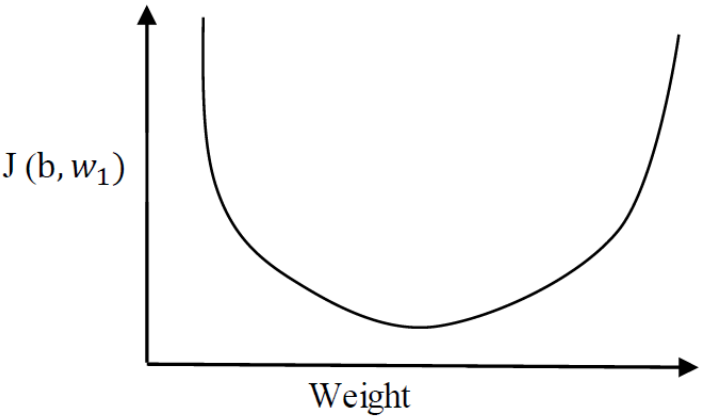

Figure: 1. Model space for single parameter using MSE

This learning can be done using gradient descent or its variation. Indeed, Mean Square Error (MSE) is sensitive to outliers. Meaning, a single outlier can create some noise in the learning process. The other model parameter is Mean Absolute Error (MAE) [11].

$$J\ (b, w_1,w_2, \ldots, w_m) = \frac{1}{n}\sum_{i=1}^{n}\left|h_{b,w_1, \ \ldots \ ,w_m}\left(x^{(i)}\right) - y^{(i)}\right| \qquad (3)$$

which if use, produces a kink shape curve at the learning stage.

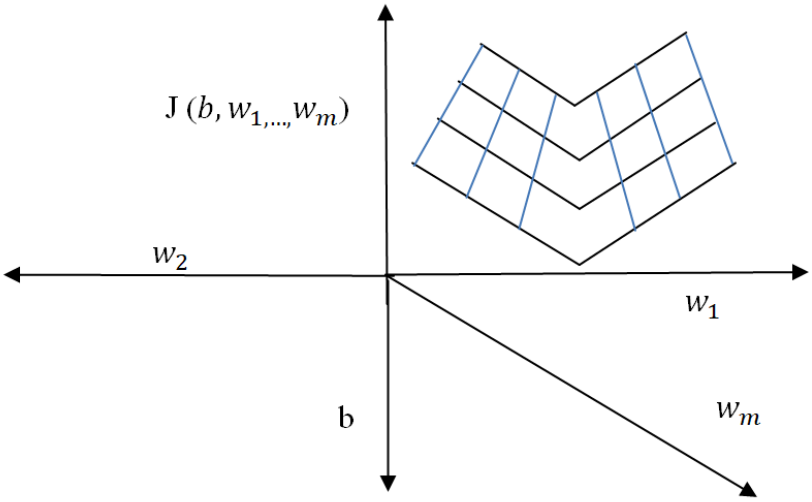

Figure: 2. Model space for multiple parameters using MAE

Mean Absolute Error (MAE) is more robust to outliers, meaning, features (inputs) are directly proportional to the target (output). When we use mean absolute error, at the global minimum, the function is non-smooth/non-differentiable.

There is the need to develop a generalized gradient method (Sub-gradient Method) that will solve non-smooth convex loss function with improved accuracy and efficiency.

1.1. Statement of the Problem

Several authors paid attention to the series of setbacks of Mean Square Error, among the authors are, [14], [4], [8] etc. They still continue using Mean Square Error (MSE) for its differentiable property. We adopted Mean Absolute Error (MAE) for its robustness/non-sensitiveness to outliers knowing fully well that at the learning stage it is not every point of it that is differentiable. Batch sub-gradient method is used to solve non-smooth convex problem but it is consuming time and memory because iteration is over the entire dataset. Stochastic sub-gradient methods consume less memory because iteration/epoch is over single data point at a time. Therefore, we propose batch-stochastic variation of sub-gradient method for computational efficiency because epoch is defined over collection of data points.

1.2. Aim and Objectives of the Study

The aim of the research is to develop a batch-stochastic sub-gradient method for the direct solution of loss function problems that are non-smooth but convex.

The specific objectives are to:

I. Unravel the mystery behind sub-gradient in conjunction to machine learning
II. Develop a batch-stochastic sub-gradient algorithm
III. Determine the efficiency of the method using numerical examples

1.3. Scope and Limitation of the Study

The study is only restricted to training loss function that are non-smooth and convex.

1.4. Significance of the Study

The field of sub-gradient is unfortunate to be blessed with several high-quality, comprehensive research-level monographs. Combining two or more sub-gradient algorithm that will result into more stable, efficient, and effective algorithm than each individual algorithm is an open research topic in optimization. Studying which area of knowledge can the combined, sub-gradient algorithm fit in correctly is also an open research area in optimization. Studying how fast sub-gradient work on specific machine learning, image processing problems, etc. is still a topic of research. There is not a good stopping criteria for the sub-gradient method. Meaning in sub-gradient a lot of things are still unravel than in many other areas of optimization.

1.5. Operational Definition of Basic Terms

1.5.1. Algorithm

Is a step by step unambiguous process that if followed correctly could solve a particular problem. Machine learning algorithms are integrated into just about every kind of device and hardware, from smartphones to servers to watches and sensors. They are increasingly the backbone behind many technological innovations and benefits, from ridesharing to autonomous vehicles to spam filtering, and many more.

Machine Learning Algorithm: Machine Learning (ML) algorithms are computer programs that adapt and evolve based on the data they process to produce predetermined outcomes. They are essentially mathematical models that “learn” by being fed data — often referred to as “training data.” Practical applications of ML algorithms includes recommendation, fraud detection and the automatic delivery of personalized marketing offers in retail. ML is the most widely used and fastest-growing subset of Artificial Intelligence today. Used to improve a wide array of computing concepts, including computer programming itself.

1.5.2. Loss Function

Measures how far an estimated value is from its true value. It is helpful in determining which model perform better and which parameters are better

Loss (MSE) $=\frac{1}{2}\sum_{i=1}^{n} \quad (h_{b,w_1, \ldots, w_m}(x^{(i)}) - y^{(i)})^2$, has differentiable property and sensitive to outliers

Loss (MAE) $=\frac{1}{n}\sum_{i=1}^{n}\left|h_{b,w_1, \ldots, w_m}(x^{(i)}) - y^{(i)}\right|$, has non differentiable property and non-sensitive to outliers

1.5.3. Weights

Decides how much influence the input will have on the output.

1.5.4. Bias

Is the effect value given to the model? Bias is used to shift the model in a particular direction. It is similar to a y-intercept. 'b' is equal to 'y' when all the features values are zero.

1.5.5. Best Fit Line

Line of best fit refers to a line through a scatter plot of data points that best expresses the relationship between those points. In finance, the line of best fit is used to identify trends or correlation in market returns between assets or over time.

1.5.6. Outliers

Are those data points that are significantly different from the rest of the dataset. They are often abnormal observation that skew the data distribution, and arise due to inconsistent data entry, or erroneous observations.

1.5.7. Structured Query Language (SQL)

Is a programming language for storing and processing information in a relational database. A relational database stores information in tabular form, with rows and columns representing different data attributes and the various relationships between the data values. You can use SQL statements to store, update, remove, search, and retrieve information from the database. One can also use SQL to maintain and optimize database performance.

2. RELATED WORKS

Most state of the art machine learning techniques revolve around the optimization of loss functions. Defining appropriate loss functions is therefore critical to successfully solving problems in this field [5]. Over past few decades, numerous machine learning algorithms have been developed for solving various problems arising in practical applications. Loss function is one of the most significant factors influencing the performance of algorithm. Many are confused about the reason why these loss functions are effective in corresponding models. The confusion further interfere them to select reasonable loss functions for their algorithms [7].

Telling a machine to figure out the best way to do something has gotten a lot of attention recently and that is Machine Learning (ML) (subfield of Artificial Intelligence (AI)). In optimization, supervised problems are part of machine learning problems which might be smooth/non-smooth, convex/nonconvex [6]; [10].

2.1. Supervised Learning

As the first branch of machine learning, supervised learning comprises learning patterns from labeled datasets and decoding the relationship between input variables (independent variables) and their known output (dependent variable). An independent variable (expressed as an uppercase “X”) is the variable that supposedly impacts the dependent variable (expressed as a lowercase “y”). For example, the supply of oil (X) impacts the cost of fuel (y).
Supervised learning works by feeding the machine sample data containing various independent variables (input) and the desired solution/dependent variable (output). The fact that both the input and output values are known qualifies the dataset as “labeled.” The algorithm then deciphers patterns that exist between the input and output values and uses this knowledge to inform further predictions.Using supervised learning, for example, we can predict the market value of a used car by analyzing other cars and the relationship between car attributes (X) such as year of make, car brand, mileage, etc., and the selling price of the car (y). Given that the supervised learning algorithm knows the final price of the cars sold, it can work backward to determine the relationship between a car ’ s value (output) and its characteristics (input).After the machine deciphers the rules and patterns between X and y , it creates a model: an algorithmic equation for producing an outcome with new data based on the underlying trends and rules learned from the training data. Once the model is refined and ready , it can be applied to the test data and trialed for accuracy. Examples of commons algorithms used for supervised learning include regression analysis (i.e. linear regression, logistic regression, non-linear regression), decision trees, k-nearest neighbors, neural networks, and support vector machines [13].

As a sub-branch of artificial intelligence, machine learning is actively developing in various areas of technology and human life. It is used in several computational problems in which the development and programming of explicit algorithms with good performance are difficult or impossible. Examples of its applications include email filtering, detecting network criminals or malicious insiders, optical character recognition, ranking training, computer vision, etc.

For so-called "training", machine learning often uses statistical techniques and algorithms, one of which is the gradient descent method [4]. Smooth convex optimization problems can be solved using gradient decent or its Variations.

2.2. Gradient Descent

Gradient descent is a first order optimization algorithm. To find a local minimum of a function using gradient descent, one takes steps proportional to the negative of the gradient (or of the approximate gradient) of the function at the current point. If instead one takes steps proportional to the positive of the gradient, one approaches a local maximum of that function; the procedure is then known as gradient ascent. Gradient descent is also known as steepestdescent, or the method of steepest descent. Gradient descent should not be confused with the method of steepest descent for approximating integrals.

Using the Gradient Decent (GD) optimization algorithm, the weights are updated incrementally after each epoch (= pass over the training dataset). The loss function J (·), the Sum of Squared Errors (SSE), can be written as:

$$\mathrm{J}\ (w) = \frac{1}{2}\sum_i (target^{(i)} - \ output^{(i)})^2 \qquad (4)$$

The magnitude and direction of the weight update is computed by taking a step in the opposite direction of the loss gradient is given by

$\Delta w_j = \ \eta \frac{\partial J}{\partial w_j}$, where η is the learning rate. The weights are then updated after each epoch via the following update rule:

W: = w + Δw, where Δw is a vector that contains the weight updates of each weight coefficient w, which are computed as follows:

$$\Delta w_j = \ -\eta \frac{\partial J}{\partial w_j} \qquad (5)$$

$$= \ -\eta \frac{\partial J}{\partial w_j} \sum_i (target^{(i)} - \ output^{(i)})\ (-x_j^{(i)})$$

$$= \ \eta \frac{\partial J}{\partial w_j} \sum_i (target^{(i)} - \ output^{(i)})\ (x_j^{(i)})$$

Essentially, we can picture GD optimization as a hiker (the weight coefficient) who wants to climb down a mountain (loss function) into a valley (loss minimum), and each step is determined by the steepness of the slope (gradient) and the leg length of the hiker (learning rate) [9].

In real life problems. Some loss functions are not differentiable (smooth) as such making sub-gradient method a generalized gradient descent methods [15].

Non-smooth convex optimization problems can be solved using sub-gradient method or variations of sub-gradient methods.

2.3. Sub-Gradient Optimization (or Sub-Gradient Method)

Is an iterative algorithm for minimizing convex functions, used predominantly in non-smooth optimization for functions that are convex but non-smooth. It is often slower than Newton's Method when applied to convex differentiable functions, but can be used on convex non-smooth functions where Newton's Method will not converge. It was first developed by Naum Z. Shor in the Soviet Union in the 1960's [12].

The Sub-gradient (related to Sub-derivative and Sub-differential) of a function is a way of generalizing or approximating the derivative of a convex function at non-smooth points. The definition of a sub-gradient is as follows: g is a sub-gradient of f at x if, for all y, the following is true: f: $R^n \rightarrow R$ is any g ϵR^n

g is a sub-gradient of f (not necessarily convex) at x if f (y) ≥ f (x) $+g^T(y - x)$, ∀y

Important Note

If f is differentiable then its tangent only touch at a singular point $(\boldsymbol{x_1}) \Rightarrow \boldsymbol{g} = \{\boldsymbol{\nabla f(x)}\}$**,** if **f** is non-differentiable (non-smooth) then there are many tangents at that particular non-differentiable point **(x_1)⇒**Sub-differential **[∂f (x)] = {g: g is a sub-gradient}** Indeed, sub-differential exist for convex or non- convex function. But it guaranteed to exist for convex function.

Sub-gradient use to come up in several contexts

i. Algorithm for non-differentiable convex optimization

ii. Convex analysis e.g., optimality conditions, duality for non-differentiable problems (f (y) ≤ f (x) +$g^T(y - x)$, ∀y, then g is a super-gradient)

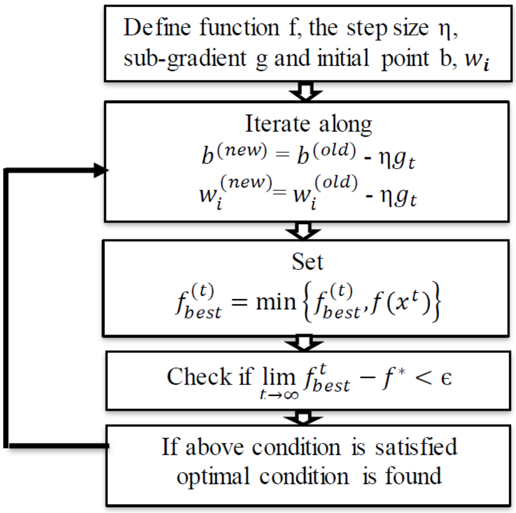
Define function f, the step size η, sub-gradient g and initial point b, w_i
Iterate along
$b^{(new)} = b^{(old)} - \eta g_t$
$w_i^{(new)} = w_i^{(old)} - \eta g_t$
Set
$f_{best}^{(t)} = \min\left\{f_{best}^{(t)}, f(x^t)\right\}$
Check if $\lim_{t\to\infty} f_{best}^{t} - f^* < \epsilon$
If above condition is satisfied optimal condition is found

Figure: 3 Algorithm flowchart for the sub-gradient method [1]

Step Size Rules [Step sizes are fixed ahead of time**]**

Several different step size rules can be used:

i. Constant step size: $\eta_k = h$ independent of k.

ii. Constant step length: $\eta_k = h/\left\|g^{(k)}\right\|_2$. This means that $\left\|x^{(k+1)} - x^{(k)}\right\|_2 = h$.

iii. Square summable but not summable: These step sizes satisfy
$\sum_{k=1}^{\infty} \eta_k^2 < \infty$, $\sum_{k=1}^{\infty} \eta_k < \infty$,
One typical example is $\eta_k = a/(b + k)$, where a > 0 and b ≥ 0.

iv. Non-summable diminishing: These step sizes satisfy
$\lim_{k\to\infty} \eta_k = 0$, $\sum_{k=1}^{\infty} \eta_k = \infty$. One typical example is$\eta_k = a/\sqrt{k}$, where a > 0.

An important thing to note is that for all four of the rules given here, the step sizes are determined "off-line", or before the method is iterated. Thus the step sizes do not depend on preceding iterations. This "off-line" property of sub-gradient methods differs from the "on-line" step size rules used for descent methods for differentiable functions where the step sizes do depend on preceding iterations.

2.3.1. Properties of Sub-Gradient Optimization

i. ∂f (x) is closed and convex set

ii. If f is convex then ∂f (x) is non empty and bounded

iii. If f is convex and differentiable, $\nabla f(x)$ is a sub-gradient of f at x
iv. If f is convex then x is a minimizer if and only if 0 ϵ ∂f (x)
v. $g$is a sub-gradient of f at x iff (g, -1) support epi f at (x, f (x))
vi. $g\ is$ an underestimator iff f (x) +$g^T(y - x)$ is a global affine under estimator (f)
vii. If f is convex, then the sub-gradient (gradient) is monotone:
< ∇f (x) - $\nabla f\ (y)$, x-y > ≥ 0
<$g_x - g_y$, x - y > ≥ 0, for g_x ϵ $\partial f(x),\ g_y$ϵ $\partial f(y)$

2.3.2. Convergence, Stability and Other Analysis of Sub-Gradient Optimization

The necessary and sufficient optimality condition:
0ϵ$\partial f(x)$ [from equation to an inclusion (differential inclusion)]
F: $R^n \rightarrow R$,$\partial f(x) \neq$ { }, for all x ϵ R^n

2.3.3. Convergence Results

There are different results on convergence for the sub-gradient method depending on the different step size rules applied. For constant step size rules and constant step length rules the sub-gradient method is guaranteed to converge within some range of the optimal value. Thus:

$$\lim_{k\to\infty} f_{\text{best}}^{(k)} - f^* < \epsilon$$

where f^* is the optimal solution to the problem and ϵ is the aforementioned range of convergence. This means that the sub-gradient method finds a point within ϵ of the optimal solutionf^*. ϵ is number, that is, a function of the step size parameter h and as h decreases the range of convergence ϵ also decreases, *i.e.* the solution of the sub-gradient method gets closer to f^* with a smaller step size parameter h. For the diminishing step size rule and the square summable but not summable rule, the algorithm is guaranteed to converge to the optimal value or$\lim_{k\to\infty} f(x^{(k)}) = f^*$. When the function f is differentiable the sub-gradient method with constant step size yields convergence to the optimal value, provided the parameter h is small enough.

2.3.4. Non-Smooth Analysis

The theory of non-smooth analysis is based on convex analysis.

2.3.5. Convex Analysis

Definition. The sub-differential of a convex function f:R^n → R at x ϵ R^n is the set ∂_cf (x) of vectors g ϵ R^n such that $\partial_c\ f(x) = \{g$ ϵ $R^n|$ F (y) ≥ f(x) +$g^T(y - x)$, ∀ y ϵ R}
Each vector g ϵ $\partial_c f(x)$ is called a sub-gradient of f at x.

Theorem 1

Let f: R^n → R be a convex function. Then for all x ϵ R^n,

$f^1(x;d)$ = max {$g^T d|g$ ϵ $\partial_c f(x)$}, ∀ d ϵ R^n and
$\partial_c f(x)$ = {g ϵ $R^n|f^I$g ϵ R^n|f^I (x, d) ≥ $g^T d$, ∀ d ϵ R^n } [3].

Example: Absolute value function

Function f(x) = |x| is clearly convex and differentiable when x is not 0. By definition of sub-differential G $\epsilon\partial_c f(0)$ = $|y| \geq |0| + g\,(y - 0)\ \forall\ y\ \epsilon\ R$

$= |y| \geq |0| + g(y)$
$= g \leq 1$ and $g \geq -1$

Thus, $\partial_c f(0)$= [-1, 1].

2.3.6. Advantages and Disadvantages of Sub-Gradient Optimization

i. Sub-gradient method has disadvantage that it can take much slower than interior point methods such as Newton's method. It has the advantage of the memory requirement being often times much smaller than those of an interior point or Newton method, which means, it can be used for extremely large problems for which interior point or Newton methods cannot be used.
ii. Gradient descent improves at every iteration, unlike sub-gradient.
iii. Gradient descent can take a "big" step size; self tunning property, unlike sub-gradient.
iv. Gradient descent guarantee improvement (Gradient descent is a descent algorithms), unlike sub-gradient.
v. The set of tangent of non-smooth functions are global under-estimator
vi. Many of the things that can be done with gradient can be extended to even non-smooth function as far as they are convex
vii. Sub-gradient is also useful for exponential functions since the rate of growth increases rapidly. On the other hand, the use of sub-gradient in optimization algorithm for some class of optimization problems may give us more efficient results.

2.3.7. Advantages and Disadvantages of Batch Sub-Gradient Method

i. Stable but computationally expensive, it is also memory in-efficientbecause it needs more memory to load all the complete training data into memory at once
ii. Fewer oscillation process and easy convergence to global minima
iii. It is less prone to local minima but in case it tends to it will not be able to come out of it due to no-noisy steps

2.3.8. Advantages and Disadvantages of Stochastic Sub-Gradient Method

i. Non-stable but computationally very fast, it is also memory efficient because it consider one observation at a time from the complete dataset.
ii. Stochastic ensure that we are not stop on local minima if it exist by searching around to get out of the local minima and move to the global minima
iii. It does not converge straight because of the noise

2.3.9. Advantages and Disadvantages of Batch-Stochastic Sub-Gradient Method

i. Stable than stochastic and computationally efficient than batch becauseit consider defined collection of observation at a time from the complete dataset.

ii. Fewer oscillation process and easy convergence to global minima

iii. It needs less memory to load defined collection of observation (data) into memory at once.

3. METHODOLOGY

3.1. Hypothesis Space: Linear functions

$$Y = b + \sum_{i=1}^{m}(w_i x_i) + e \qquad (6)$$

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, X = \begin{bmatrix} 1 & x_{11} & x_{21} & \cdots & x_{n1} \\ 1 & x_{12} & x_{22} & \cdots & x_{n2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{1m} & x_{2m} & \cdots & x_{nm} \end{bmatrix}, \beta = \begin{bmatrix} b \\ w_1 \\ \vdots \\ w_n \end{bmatrix} \text{ and } e = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix}$$

3.2. Loss Function

$$J\,(b, w_1,w_2, \ldots, w_m) = \frac{1}{n}\sum_{i=1}^{n}\left|h_{b,w_1, \ldots ,w_m}\left(x^{(i)}\right) - y^{(i)}\right| \quad (7)$$

3.3. Optimization/ Learning Problem

Find out b, $w_1,w_2, \ldots, w_m$ such that J (b, $w_1,w_2, \ldots, w_m$) is minimized

3.3.1. Proposed Algorithm [Batch-Stochastic Sub-Gradient Algorithm]

Suppose 1 < p < q; say p = 1,000,000, q =10, p/q = 100,000 batches

```
I.     Load the first batch of j
II.    Randomly initialized b, $w_1, w_2, \ldots, w_m$ for the loaded record
III.   Repeat until convergence
IV.    Predict $\check{y}^{(i)}$for the batch of j in training
V.     Calculate loss $J(b, w_1, w_2, \ldots, w_m)$
VI.    Pick $g_t$ such that f (y) ≥ f (($w_t$) + $g_t^T(y - w_t)$
VII.   $b^{(new)} = b^{(old)}$ - η$g_t(b)$
           $w_1^{(new)}$= $w_1^{(old)}$ - η$g_t(w_1)$
           .
           .
           .
           $w_m^{(new)}$= $w_m^{(old)}$ - η$g_t(w_m)$
           Update$b, w_1, w_2, \ldots, w_m$ Simultaneously
           Load j = new j
VIII.      Go to step III
```

3.4. Data Typed Used

Inputted testing data

4. RESULTS

4.1. Numerical Example for Single Parameter

Suppose we want to make an algorithm for real estate that can estimate the market price of a house. First we need some data from which the algorithm can learn from.

Table 1. House prices in square feet (inputted testing data)

House (square feet) [X_1]	Price [y]
1000	150,000
2000	400,000
3000	550,000
4000	625,000
5000	825,000

If we analyze the data in table 1 using SQL, we will have

Table 2. SQL analysis for house prices

R square	Standard error value of y	Degree of freedom	SS regression	SS residual
0.9699	50621.1418	3	2.48063E+11	7687500000

Setting up our Learning Environment:

4.1.1. Batch sub-Gradient Method

Table 3. Best batch sub-gradient capture

b	w_1	House (square feet)	Price	h(x)	|h(x)-y|
37500	157.5	1000	150000	195000	45000
		2000	400000	352500	47500
		3000	550000	510000	40000
		4000	625000	667500	42500
		5000	825000	825000	0

Table 4. Batch sub-gradient iteration

Iteration	b	w_1	MAE
I	37500	157.5	35000
I+1	37499.99	157.49	35010.002
I+2	37499.98	157.48	35020.004

For that predicted,

$\breve{\boldsymbol{y}}^{(\boldsymbol{i})}$ = 37500X_0 + 157.5 X_1 (8)

Table 5. Best batch sub-gradient analysis (iteration I)

R square	Standard error value of y	Degree of freedom	Sum of h(x)	Sum of errors
1	50621.1418	3	2550000	0

4.1.2. Stochastic Sub-Gradient Method

Table 6. Stochastic sub-gradient capture

B	w_1	House (square feet)	Price	h(x)	|h(x)-y|
37500	157.5	1000	150000	195000	45000
37499.99	157.49	2000	400000	352500	47500
37499.98	157.48	3000	550000	510000	40000
37499.97	157.47	4000	625000	667500	42500
37499.96	157.46	5000	825000	825000	0

Table 7. Stochastic sub-gradient iteration

Iteration	B	w_1	MAE
J	37500	157.5	35000
J+1	37499.96	157.46	35040.008
J+2	37499.92	157.42	35080.016

For that predicted,

$\breve{\boldsymbol{y}}^{(\boldsymbol{i})}$ = $37500X_0 + 157.5X_1$ (9)

Table 8. Best stochastic sub-gradient analysis (iteration J)

R square	Standard error value of y	Degree of freedom	Sum of h(x)	Sum of errors
1	3.76E-11	3	2550000	0

4.1.3. Batch-Stochastic Sub-Gradient Method

Table 9. Batch-stochastic sub-gradient capture

b	w_1	House (square feet)	Price	h(x)	|h(x)-y|
37500	157.5	1000	150000	195000	45000
		2000	400000	352500	47500
		3000	550000	510000	40000
37499.99	157.49	4000	625000	667500	42500
		5000	825000	825000	0

Table 10. Batch-stochastic sub-gradient iterations

Iteration	B	w_1	MAE
K	37500	157.5	35000
K+1	37499.98	157.48	35020.004
K+2	37499.96	157.46	35040.008

For that predicted

$$\breve{\boldsymbol{y}}^{(\boldsymbol{i})} = 37500X_0 + 157.5X_1 \qquad (10)$$

Table 11. Best batch-stochastic sub-gradient analysis (iteration K)

R square	Standard error value of y	Degree of freedom	Sum of h(x)	Sum of errors
1	3.76E-11	3	2550000	0

Comparison of SQL analysis, best batch sub-gradient analysis best stochastic sub-gradient analysis and best batch-stochastic sub-gradient analysis

Table 12. Comparison for best house prices analysis

Analysis tools	**SQL Analysis**	**Batch**	**Stochastic**	**Batch-Stochastic**
R squared	0.9699	1	1	1
Standard error value of y	50621.1418	3.76E-11	3.76E-11	3.76E-11
Degree of freedom	3	3	3	3
Sum of y	2550000	-	-	-

Sum of h(x)	-	2550000	2550000	2550000
Sum of errors	-	0	0	0
MAE	-	35000	35000	35000

4.2. Numerical Example for Multiple Parameters (Systolic Disease)

Numerical Example for Multiple Parameters (Systolic Disease)

Suppose we have X_1 as Age X_2 as Weight X_3 as Parents and y as intensity of Systolic disease on infected person based onX_1,X_2, andX_3 values.

Table 13. Systolic disease (inputted testing data)

Age	Weight	Parents	Systolic
54	194	1	141
40	206	1	131
47	221	0	147
53	200	0	134
57	209	1	144
47	212	0	140
54	177	2	132
48	202	0	138
46	185	0	120
50	199	2	137
53	229	2	151
54	203	2	145
61	207	2	139
43	215	1	144
53	214	1	140
53	202	1	142
53	228	1	145
42	196	1	130
52	199	0	130
48	216	1	146
46	228	1	144
63	199	2	150
39	203	1	133
51	226	1	142
50	207	1	138

If we analyze the data in Table 13 using SQL, we will have

Table 14. SQL analysis of the Systolic data

R square	Standard error value of y	Degree of freedom	SS regression	SS residual
0.702	4.209	21	877.478	371.962

Setting up our Learning Environment:

4.2.1. Batch Sub-Gradient Method

Table 15. Systolic batch sub-gradient iteration

Iteration	b	w_1	w_2	w_3	MAE
I+6	34.45665	2.025309	0.064945	0.127167	12.3605157
I+7	34.44665	2.015309	0.054945	0.117167	11.13651439
I+8	34.43665	2.005309	0.044945	0.107167	10.27438283
I+9	34.42665	1.995309	0.034945	0.097167	9.775675174
I+10	34.41665	1.985309	0.024945	0.087167	9.450457783
I+11	34.40665	1.975309	0.014945	0.077167	9.460488391
I+12	34.39665	1.965309	0.004945	0.067167	10.06070509
I+13	34.38665	1.955309	-0.00506	0.057167	11.05376309
I+114	34.37665	1.945309	-0.01506	0.047167	12.2546383

The model so far has learned and that predicted,

$$\breve{\boldsymbol{y}}^{(\boldsymbol{i})} = 34.41665X_0 + 1.985309X_1 + 0.024945X_2 + 0.087167X_3 \qquad (11)$$

Table 16. Best systolic batch sub-gradient analysis (iteration I+10)

R square	Standard error value of y	Degree of freedom	Sum of h(x)	Sum of error
1	1.27E-14	21	3487.269	4.269

4.2.2. Stochastic Sub-Gradient Algorithm

Table 17. Systolic stochastic sub-gradient iteration

Iteration	b	w_1	w_2	w_3	MAE
J+5	34.77665	2.345309	0.384945	0.447167	93.23930761
J+7	34.52665	2.095309	0.134945	0.197167	28.60887283
J+8	34.27665	1.845309	-0.11506	-0.05283	36.02156196
J+9	34.27665	1.845309	-0.11506	-0.05283	36.02156196
J+10	34.02665	1.595309	-0.36506	-0.30283	100.6519967

The model so far has learned and that predicted,

$$\breve{\boldsymbol{y}}^{(\boldsymbol{i})} = 34.52665X_0 + 2.095309X_1 + 0.134945X_2 + 0.197167X_3 \qquad (12)$$

Table 18. Best systolic stochastic sub-gradient analysis (iteration J+7)

R square	Standard error value of y	Degree of freedom	Sum of h(x)	Sum of error
1	1.36E-14	21	4200.509	717.5091

4.2.3. Batch-Stochastic Sub-Gradient Method

Table 19. Systolic batch-stochastic sub-gradient iteration

Iteration	b	w_1	w_2	w_3	MAE
K+2	34.49665	2.065309	0.104945	0.267167	20.95322065
K+3	34.48665	2.055309	0.094945	0.257167	18.40841378
K+4	34.46665	2.045309	0.084945	0.247167	16.15930917
K+4	34.46665	2.035309	0.074945	0.237167	14.03939457
K+5	34.45665	2.025309	0.064945	0.227167	12.41703743
K+6	34.44665	2.015309	0.054945	0.217167	11.19095048
K+7	34.43665	2.005309	0.044945	0.207167	10.32220891
K+8	34.42665	1.995309	0.034945	0.197167	9.806109957
K+9	34.41665	1.985309	0.024945	0.187167	9.480892565
K+10	34.40665	1.975309	0.014945	0.177167	9.447444913
K+11	34.39665	1.965309	0.004945	0.167167	10.0302703
K+12	34.38665	1.955309	-0.00506	0.157167	11.01463265
K+13	34.37665	1.945309	-0.01506	0.147167	12.20681222
K+14	34.36665	1.935309	-0.02506	0.137167	13.76918474

The model has so far learned and that predicted

$$\breve{\boldsymbol{y}}^{(\boldsymbol{i})} = 34.40665X_0 + 1.975309X_1 + 0.014945X_2 + 0.177167X_3(13)$$

Table 20. Best systolic batch-stochastic sub-gradient analysis (iteration K+10)

R square	Standard error value of y	Degree of freedom	Sum of h(x)	Sum of error
1	1.14E-14	21	3424.929	-58.0709

Table 21. Comparison of systolic analysis

Analysis tools	**SQL Analysis**	**Batch**	**Stochastic**	**Batch-Stochastic**
R squared	0.7023	1	1	1
standard error value of y	4.2086	1.27E-14	1.36E-14	1.14E-14

Degree of freedom	21	21	21	21
Sum of y	3483	-	-	-
Sum of h(x)	-	3487.269	4200.509	3424.929
Sum of errors	-	4.269	717.509	-58.071
MAE	-	9.45046	28.60887	9.44744

5. DISCUSSIONS OF FINDINGS

Table 1 shows the size of house (square feet) and its corresponding price. Table 2 shows 96.994% co-efficient of determination which means people are buying house (square feet) with about 3.1% inaccuracy in the paid price.

Column h(x) in table 3, 6 and 9 shows the predicted price for the corresponding house (square feet) with 100% accuracy for both batch, stochastic and batch-stochastic sub-gradient method.

Table 4, 7 and 10 shows the highest convergence of the three methods from there the MAE will start diverging. Table 5, 8 and 11 shows iteration I+1, J+1 and K+1 has the lowest MAE with best coefficient of the independent variable. Table 11 shows the comparison of the three method. Table 13 shows the intensity of systolic disease on infected person based on age, weight and parental history of the said disease. Table 14 shows about 70.229% coefficient of determination which means there is about 29.771% inaccuracy in the observation.

Table 15 row (I+10) has the lowest MAE value (9.4505) with the best independent variables, table 16 shows that using batch method the highest convergence it can goes from there it will start diverging.

Table 17 row (J+7) has the lowest MAEvalue (28.6089) with the best independent variables, table 18 shows that using stochastic method the highest convergence it can goes from there it will start diverging.

Table 19 row (K+10) has the lowest MAE value (9.4474) with the best independent variables, table 20 shows that using batch-stochastic method the highest convergence it can goes from there it will start diverging.

Table 21 shows the comparison of the tree methods based on the values of independent variables with 100% accuracy for both batch, stochastic and batch-stochastic sub-gradient method.

Table 20 and 21 shows batch-stochastic sub-gradient method is the best for making predictions on a continuous data because we are able to split the data into different warehouse and also batch-stochastic MAE value (9.4474) is more close to batch MAE value (9.4505) than stochastic MAE value (28.6089). Meaning our record has less memory requirement than batch and more stability than stochastic.

6. SUMMARY, CONCLUSION AND RECOMMENDATION

6.1. Summary

Mean absolute Error (MAE) is one of the first discovered L1 loss functions but has non-differentiable property. Scholars used to point out its advantages but continue to use one of the second discovered L2 loss functions (i.e., Mean Square Error (MSE) which has differentiable property. To researchers that have delve in knowledge of loss functions knows that MSE attract noises for that root was added (i.e., RMSE (Root Mean Square Error)) to penalized the noises, but to us effectiveness of RMSE is still far below from that of MAE.

With the trainings and experience over the years, I adopted MAE and with one go batch-stochastic sub-gradient method was developed and I have tested the stability and computational efficiency of the new method using Structured Query Language (SQL).

6.2. Conclusion

Of all the machine learning loss functions; mean absolute error, mean square error, root mean square error, mean absolute percentage error, mean square logarithmic error, squared hinge, categorical hinge, soft pairwise hinge, pairwise logistic, logcosh, categorical cross entropy, sparse categorical cross entropy, binary cross entropy, kullbackleibler divergence, gaussian negative log likelihood, huber, poisson, triple margin, soft margin, margin ranking, cosine proximity etc., only Mean Absolute Error (MAE), Mean Square Error(MSE) and Root Mean Square Error (RMSE) can fit well into training a continuous data. No any other formal gimmicks can one apply on a continuous data that will give the most accurate prediction other than batch-stochastic sub-gradient method.

6.3. Recommendations

We recommend development of high-quality comprehensive research level monograph on Mean Absolute Error (MAE), Mean Square Error (MSE), Root Mean Square Error(RMSE), batch sub-gradient method and stochastic sub-gradient method. We also recommend having good grasp of how Structured Query Language (SQL) works. Lastly, we recommend choosing small learning rate especially when we are near optimal point.

6.4. Areas for Further Research

Future research can consider implementation of batch-stochastic sub-gradient method as software package and deployment of the package to neural networking, controls, communication, economics, rainfall analysisetc. also, future research can consider development of stable and effective method for solving non-smooth, non-convex problems.

6.5. Contribution to Knowledge

Batch-stochastic sub-gradient method with better computing time and less memory requirement than batch sub-gradient method and more stability than stochastic sub-gradient method has been developed. The new method shows greater stability, efficiency, accuracy and convergence than any other existing method of minimizing a continuous data with single or multiple parameter(s).

ACKNOWLEDGEMENTS

Praise be to the eternally besought of all. The all-Knowing (al-Alim), the all-Powerful (al-Qadir); whose power controls every single soul and whose will dominates every single event.

My profound gratitude goes to Prof. M. R. Odekunle, Prof. Y. D. Mamman, Prof. I. I. Adamu, Prof. S. Musa, Prof. A. Tahir, Prof. B. A. Madu, Dr. A. A. Momoh, Dr. Y. B. Chukkol Dr. A. M. Alkali, Dr. S. O. Adee, , Dr. H. B. Aliyu, Dr. A. Umar, Mr. P. A Atsen, Mr. U. Suleiman, Mr. M. Ishiyaku, Mr. M. Abdullahi, Mr. U. A. Abdullahi, Mr. A. Alhassan, Mr. M. Usman, Mr. A. S. Ahijo and Mr. Dahiru for their contributions, support and encouragement towards the successful completion of this work and salutation to all people in whom I see wisdom and knowledge.

I am also grateful to my friends; Abubakar Iliyasu, Bashir Saidu Musa, ShuaibuLawanKada, Shuaibu Nasidilawan, Solomon Ayuba, JamiluIlyasu, Binyaminu I. Dahiru, Ahmad M. Goji, Muh’d Bello, Bello Mustafa, Muh’d Usman, Umar A. Madaki, AnasAdamu, Muh’d Hashidu, Shuaibu Muh’d, Abdul-basidHaruna, Abdul-rashid Ishiyaku, Ibrahim Ishiyaku Kachalla, Muh’d Salisu, Kamaludden Sa’ad, HarunaMuh’d, Muhd Abdul-samad, Ibrahim Muh’d, Lukman Bello, Mahmud Muh’d, Nuhu A. Nuhu, Umar M. Bello, Abbas Ahmad, Haruna Magaji, Ahmad Abdullahi, Abubakar Haruna, Kamaluddeen Saad etc., for their support, tolerance and understanding.

To my father, mother and siblings, my struggles, heart and tears eternally yours.

REFERENCES

[1] Aaron, A. (2015). Subgradient optimization. ChE 345 spring 2015. Last edited on 1 April 2022 at 11:23. Downloaded from https://optimization.cbe.cornel.edu/ index.php/title= Subgradient _ optimization on Date: 05/02/2023. Time: 02: 13 AM

[2] Ahmadi, A. A. (2016). Basic notation and terminology in optimization. *ORF523 Lecture 3*Spring, Princeton University. Downloaded from http://www.princeton.edu/~aaa/Public/ Teaching/ORF523/S16/ORF523_S16_Lec3_gh.pdf on Date: 21/06/2022. Time: 02: 15 AM.

[3] Bagirov, A., Karmitsa, N. &Makela, M. M. (2014). Introduction to non-smooth optimization. Theory, Practice and Software Department of Mathematics and Statistics University of Turku, Finland.

[4] Chai, T. &Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)? Arguments against avoiding RMSE in the literature Cooperative Institute for Climate and Satellites, University of Mary land, College Park, MD20740, USA.

[5] Ciampiconi, L., Elwood, A., Leonardi, M., Muhamed A. &Rozza A. (2023). A survey of taxonomy of loss functions in machine learning. Lastminutes.com group, Switzerland. Cite as: arXiv:1811.00980 [math.OC] (or arXiv:1811.00980v2 [math.OC] for this version). Downloaded from https://www.doi.org/10.48550/arXiv.1811.00980 on Date: 30/09/2021. Time: 06: 17 AM.

[6] Khan, M. & Noor, S. (n.d.). Performance analysis of regression - Machine Learning Algorithms for Predication of Runoff Time. Department of Electronic and Computer Science, University of Southampton, UK, Research article. Downloaded from https://www. researchgate.net/ publication/333640679_The Secrets_of_Machine_Learning_Ten_ThingsYou_Wish_You_ Had Known Earlier_to_be_More_Effective_ at_Data_Analysis/ link/5cf88930299bf1fb185bb bab/ download on Date: 12/01/2021. Time: 09: 11 AM.

[7] Nie, Z., Hu, Z. & Li, X. (2018). An investigation for loss functions widely used in machine learning. Communication in information and systems volume 18, Number 1, 37-52, 2018.

[8] Padhma, M. (2022). End-to-End Introduction to Evaluating Regression Models. Data science blogathon. Downloaded from https://www.analyticsvidhya.com/ blog/2021/10/ evaluation-metric-for-regression-models on Date: 3/04/2023. Time 01:12 AM.

[9] Qadri, M. (n.d.). Stochastic Gradient Descent Algorithms and its Tuning. Downloaded from https://www.scribd.com/document/311549379/Stochastic-Gradient-Descent-Term-Paper on Date: 5/06/2023. Time: 02:07 AM.

[10] Sarker, I. H. (2021). Machine learning: Algorithms, Real world applications and research directions. Published online under exclusive license to springer nature Singapore Pre Ltd. https://www.link.springer.com/article/10.1007/s42979-021-00592-x

[11] Shah, S. (2023). Loss function is no rocket science! This article was published as part of the data science Blogathon. Downloaded from https://www.analyticsvidhya.com/ blog/2021/02/ loss-function-is-no-rocket-science on Date: 15/04/2023. Time: 02:31 AM.

[12] Shor, N. Z. (2012). Minimization methods for non-differentiable functions. Vol3. Springer Science & Business Media.

[13] Theobald, O. (2017). Machine learning for absolute beginners: A Plain English Introduction (second edition).Downloaded from https://www.amazon.com/Machine-Learning-Abslute-beginners-Introduction/dp/152095140Xon Date: 5/06/2021 Time: 02:11AM

[14] Thin, J. (2003). Continuous nowhere differentiable functions. Master thesis. Supervisor: Lech Maligranda. Downloaded from http://www.diva-portal.org/smash/get/diva2:1022983/ FULLTEXT01.pdf on Date: 5/06/2023 Time: 02:11AM

[15] Willmott, C. J. & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance Center for Climatic Research, Department of Geography, University of Delaware. Newark, Delaware 19716, USA.

AUTHOR

I KasimuJuma Ahmed obtained M.Sc. Mathematics [Computational Optimization (2023)] from ModibboAdama University Yola, Nigeria and had B. Sc. Mathematics (2015) from Federal University Kashere, Gombe State, Nigeria. I was a Higher Aeronautical Communication Officer in Lawanti International Airport, Gombe State and Currently Lecturer in Federal Polytechnic Bali, Taraba State, Nigeria. I like to always work in growing organization where creativity is encouraged and critical thinking takes center stage.