ANALYSIS OF GRADIENT DECREASE ALGORITHMS

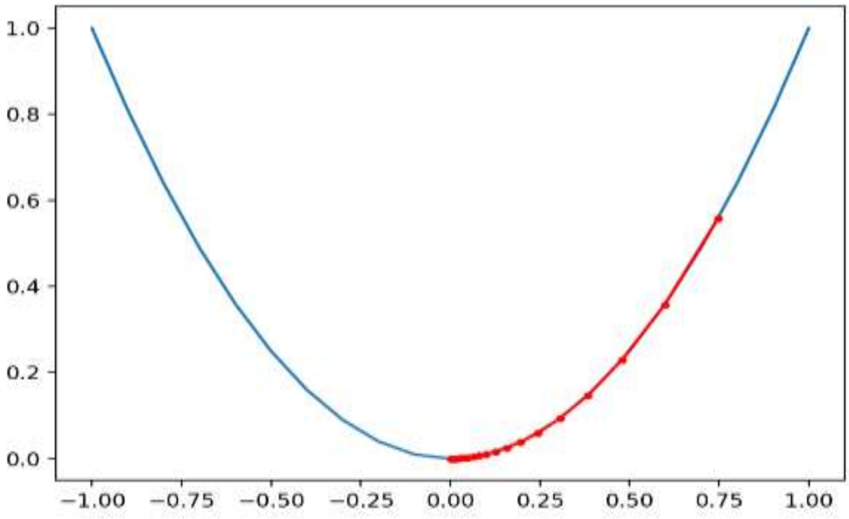
Alimardonov Shokhrukh Erkin o'g'li
Military Institute of Information and Communication Technologies and Communications of the Ministry of Defense of the Republic of Uzbekistan
shohruhalimardonov1202@gmail.com

Abstract: *This paper presents a comprehensive comparison of three gradient descent algorithms commonly used in machine learning and deep learning: Batch gradient descent, stochastic gradient descent, and mini-batch gradient descent. Differences between these algorithms in terms of gradient descent computational efficiency, stability, and learning dynamics are explained. Provides a clear and concise overview of each algorithm, their advantages and disadvantages, making it easy to understand their suitability for a specific problem and data set. Relevant information is provided to illustrate the difference between these algorithms.*

Keywords: *Gradient descent, Batch gradient, Stochastic gradient descent, Mini-Batch gradient descent.*

Introduction:

Gradient descent is an iterative algorithm for finding the minimum of a function. Its purpose is to apply optimization to find the minimum or global minimum error value. It is mainly used to update model parameters. Gradient descent is a vector-valued function that describes the slope of the tangent to the graph of the function, indicating the direction of the most significant rate of descent of the function. The main goal of this algorithm is to minimize the function by iteratively adjusting the input parameters [2, 4]. below. - we can see the gradient reduction graph in the 1.1. picture [8].



1.1 - picture. Gradient descent graphics

Batch Gradient descent formula that updates the weight parameter w:

$$w_{i+1} = w_i - a * \nabla_{w_i} J(w_i) i$$

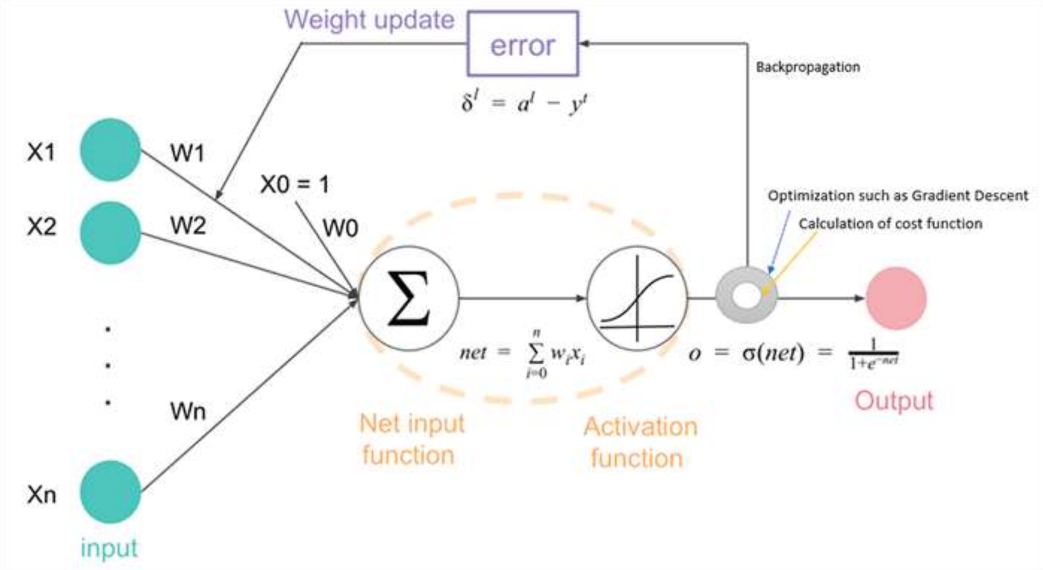
Here, w denotes the weights to be updated. a- learning speed of the algorithm. A hyperparameter that controls how much the model changes in response to an assumed error, the cost function J, also refers to the iteration index [3].

In the batch gradient descent algorithm, the entire data set is used to calculate the gradient of the cost function. 2.1 - we can see formulas

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_{predicted})^2$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - (mx_i + b)^2$$

gradient descent is computationally expensive and requires loading the entire data set into memory and evaluating the cost function for each instance [1]. Batch gradient descent computes the gradient using all the training data at each step [8].


Weight update
error
$\delta^l = a^l - y^l$
Backpropagation
X1
W1
X2
W2
X0 = 1
W0
Wn
Xn
input
Σ
$net = \sum_{i=0}^{n} w_i x_i$
Net input function
Activation function
Optimization such as Gradient Descent
Calculation of cost function
$o = \sigma(net) = \frac{1}{1+e^{-net}}$
Output

1.2. - a picture. Batch gradient descent (BGD) algorithm performance architecture.

Batch Gradient Descent (BGD) algorithm considers all training examples in each iteration. If the dataset contains a large number of training examples and a large number of features, the implementation of the Batch Gradient Descent (BGD) algorithm is computationally expensive [5].

Number of training examples to replicate = 1 million = 1^{06}

Number of iterations = 1000 = 1^{03}

Parameters must be numbered = 10000 = 1^{04}

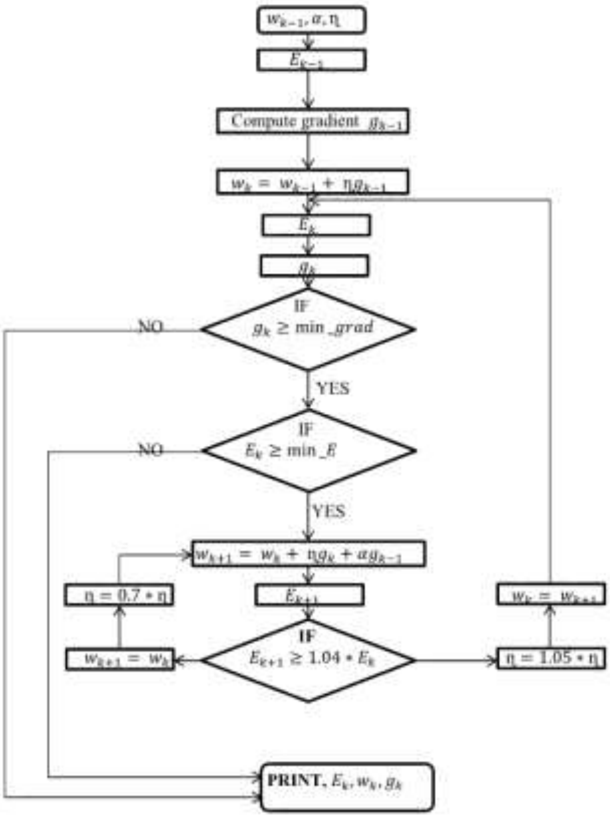
Total counts = 1^{06} * 1^{03}* 1^{04} = 1^{013}

Stochastic gradient descent, unlike batch gradient descent, does not go through all samples; Instead, it selects a single random parameter and optimizes the data set according to the recorded value of the data points, performing learning at each update [7].

A Stochastic Gradient Descent formula that updates the weight parameter w :

$$w_{i+1} = w_i - a * \nabla_{w_i} J(x^i, y^i, w_i)$$

y - Same as Gradient descent when characters are targets, and in this case represent a single observation.



1.3. - a picture. Performance architecture of stochastic gradient descent (SGD) algorithm.

Stochastic gradient descent (SGD):

Number of training examples to replicate = 1

Number of iterations = 1000 = 1^{03}

Number of parameters to be trained = 10000 = 1^{04}

Total counts = 1 * 1^{03}*1^{04}=1^{07}

Comparison with batch gradient descent:

Total accounts in BGD = 1^{013}

Total accounts in SGD = 1^{07}

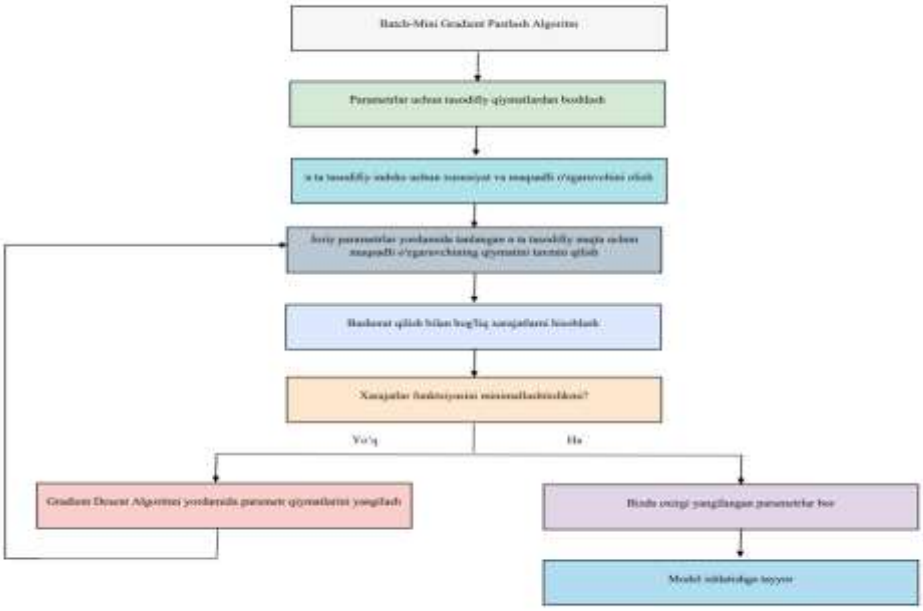
Estimate: SGD is ^{106} times faster than BGD in this example.

Mini-Batch Gradient Descent is a cross between Batch Gradient Descent (GD) and Stochastic Gradient Descent (SGD). In this approach, instead of iterating over an entire dataset or a single observation, we divide into small subsets and calculate the gradients for each subset [9].

A Stochastic Gradient Descent formula that updates the weight parameter w:

$$w_{i+1} = w_i - a * \nabla_{w_i} J(x^{i:i+b}, y^{i:i+b}; w_i)$$

The notation is the same as for Stochastic Gradient Descent, where b is a hyperparameter representing the size of one set [4].


Batch-Mini Gradient Pastlash Algoritmi
Parametrlar uchun tasodifiy qiymatlardan boshlash
n ta tasodifiy namuna uchun xususiyat va maqsadli o'zgaruvchini olish
Joriy parametrlar yordamida tanlangan n ta tasodifiy nuqta uchun maqsadli o'zgaruvchining qiymatini taxmin qilish
Bashorat qilish bilan bog'liq xarajatlarni hisoblash
Xarajatlar funktsiyasini minimallashtirdikmi?
Yo'q
Ha
Gradient Descent Algoritmi yordamida parametr qiymatlarini yangilash
Bizda oxirgi yangilangan parametrlar bor
Model ishlatishga tayyor

.4. – Performance architecture of the Mini-Batch Gradient Descent (MBGD) algorithm.

Mini-Batch gradient descent (MBGD):

Number of training examples to repeat = 100 = 1⁰²

Here we look at examples of 1⁰⁶ to 1⁰² training.

Number of iterations = 1000 = 1⁰³

Number of parameters to be trained = 10000 = 1⁰⁴

Total counts = 1⁰²*1⁰³*1⁰⁴=1⁰⁹

1.1. – table. Comparative analysis results of gradient descent methods..

N	The method name	Accuracy	Time
1	Batch gradient descent	High	More
2	Stochastic gradient descent	Low	Less
3	Mini-Batch gradient descent	Average	Average

CONCLUSION

The choice of gradient descent algorithms depends on the specific problem, the size of the data set, and the available computing resources. Batch gradient descent is suitable for small data sets and optimization problems. Stochastic gradient descent is efficient for large datasets, but requires careful tuning of the learning rate. Mini-Batch gradient descent is a popular choice for many datasets, striking a balance between accuracy and efficiency. By understanding the characteristics of each algorithm, practitioners can make decisions to optimize model training and achieve better results. We have considered the advantages, disadvantages and differences of three main types of gradient descent algorithms.

REFERENCES:

1. H. Zaynidinov, O. Mallayev, Parallel algorithm for calculating the learning processes of an artificial neural network. AIP Conference Proceedings 2647, 050006 (2022). doi: https://doi.org/10.1063/5.0104178

2. Yusupov I, Nurmurodov J, Ibragimov S, Gofurjonov M, Qobilov S. “Calculation of Spectral Coefficients of Signals on the Basis of Haar by the Method of Machine Learning”, 14th International Conference, IHCI 2022, Tashkent, Uzbekistan, October 20–22, 2022, pp 547–558. https://link.springer.com/conference/ihci

3. Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In Proceedings of COMPSTAT'2010, Springer.

4. Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

5. Ruder, S. (2016). An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747.

6. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.

7. Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: Data mining, inference, and prediction. Springer.

8. Shukla, P. (2019). The Gradient Descent Algorithm and Its Variants. arXiv preprint arXiv:1908.10448. doi: 10.1093/ptep/ptaa104

9. https://www.baeldung.com/cs/gradient-stochastic-and-mini-batch