**Experiment :- 07**

|  |
| --- |
| **Title: Write a program to implement basic Gradient Descent Algorithm** |

# Objective:

# Books/ Journals/ Websites referred:

* Markov Decision Processes in Artificial Intelligence MDPs, Beyond MDPs and Applications, Edited by Olivier Sigaud, Olivier Buffet, Wiley Publications, 2010
* <https://realpython.com/gradient-descent-algorithm-python/>
* <https://en.wikipedia.org/wiki/Gradient_descent>
* https://towardsdatascience.com/gradient-descent-algorithm-and-its-variants-10f652806a3

# Resources used:

# Theory (Students should write):

# Gradient descent is an optimization algorithm used in machine learning and other optimization problems to minimize a function. In machine learning, gradient descent is commonly used to update the parameters of a model to minimize the loss function, which measures the difference between the predicted values and the actual values.

# The basic idea of gradient descent is to iteratively update the parameters of a function in the direction of the negative gradient of the function. The negative gradient points in the direction of steepest descent, which is the direction that the function decreases the most rapidly. By taking steps in this direction, we can reach the minimum of the function.

# The gradient descent algorithm starts with an initial guess for the parameters and iteratively updates the parameters until the convergence criteria is met. The convergence criteria is usually based on the change in the loss function or the gradient of the function.

# The update rule for the parameters in gradient descent is given by:

# θ = θ - α \* ∇J(θ)

# where θ is the parameter vector, α is the learning rate, J(θ) is the loss function, and ∇J(θ) is the gradient of the loss function with respect to the parameters.

# The learning rate, α, is a hyperparameter that determines the step size of the parameter update. If α is too small, the algorithm may converge slowly, whereas if α is too large, the algorithm may overshoot the minimum and fail to converge.

# There are several variations of gradient descent, such as stochastic gradient descent and mini-batch gradient descent, which use subsets of the data to update the parameters at each iteration. These variations can be more efficient than the standard gradient descent algorithm for large datasets or high-dimensional parameter spaces.

# Algorithm (Students should write)

The gradient descent algorithm can be summarized in the following steps:

1) Initialize the parameters: Set the initial values for the parameters of the model.

2) Calculate the gradient: Compute the gradient of the loss function with respect to the parameters.

3) Update the parameters: Update the parameters using the gradient and the learning rate.

4) Repeat steps 2-3: Iterate the process of computing the gradient and updating the parameters until convergence is achieved or the maximum number of iterations is reached.

5) Output the optimized parameters: Once convergence is achieved, output the final values of the parameters that minimize the loss function.

The steps involved in the gradient descent algorithm can be described mathematically as follows:

1) Initialize the parameters:

θ = [θ\_1, θ\_2, ..., θ\_n]

where n is the number of parameters and θ\_i is the initial value of the i-th parameter.

2) Calculate the gradient:

∇J(θ) = [ ∂J(θ)/∂θ\_1, ∂J(θ)/∂θ\_2, ..., ∂J(θ)/∂θ\_n ]

where J(θ) is the loss function and ∂J(θ)/∂θ\_i is the partial derivative of the loss function with respect to the i-th parameter.

3) Update the parameters:

θ\_i = θ\_i - α \* ∂J(θ)/∂θ\_i

where α is the learning rate and ∂J(θ)/∂θ\_i is the i-th component of the gradient.

4) Repeat steps 2-3:

Iterate the process of computing the gradient and updating the parameters until convergence is achieved or the maximum number of iterations is reached.

5) Output the optimized parameters:

Once convergence is achieved, output the final values of the parameters that minimize the loss function.

# Implementation (Code):

# Kindly find the code and output in E7\_Gradient\_Descent.ipynb

# Output Screenshots:

# Conclusion (Students should write in their own words):

# gradient descent is a powerful and widely used algorithm in machine learning and optimization, and understanding its theory is essential for developing and applying machine learning models.

# Applications:

# Gradient descent is a widely used optimization algorithm in various fields, including machine learning, artificial intelligence, optimization, engineering, physics, and economics. Some common applications of gradient descent are:

# Machine Learning: Gradient descent is widely used in machine learning for training neural networks, logistic regression, support vector machines, and other models. By minimizing the loss function, gradient descent helps to optimize the parameters of the model and improve its performance.

# Image Processing: Gradient descent is used in image processing applications for edge detection, image restoration, image segmentation, and other tasks. The gradient of the image is computed using the Sobel operator or other operators, and the gradient descent algorithm is applied to optimize the image parameters.

# Robotics: Gradient descent is used in robotics for trajectory optimization, robot control, and other tasks. By minimizing the cost function, gradient descent helps to optimize the control parameters of the robot and achieve better performance.

# Natural Language Processing: Gradient descent is used in natural language processing for language modeling, machine translation, and other tasks. By minimizing the loss function, gradient descent helps to optimize the parameters of the model and improve its accuracy.

# Finance: Gradient descent is used in finance for portfolio optimization, risk management, and other tasks. By minimizing the objective function, gradient descent helps to optimize the portfolio weights and minimize the risk.

# Overall, gradient descent is a powerful optimization algorithm with numerous applications in various fields. Its ability to optimize complex functions makes it an essential tool for solving optimization problems and developing advanced machine learning models.