Name: Ayushi Singh

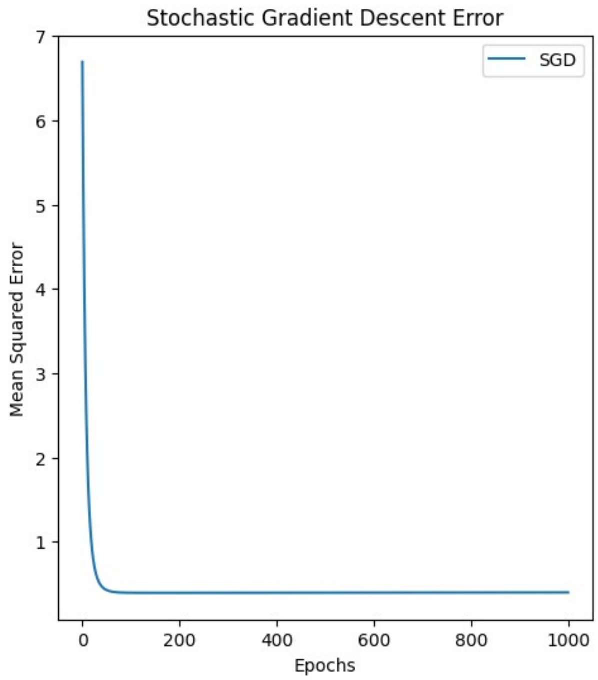
Sap Id: 60009220202

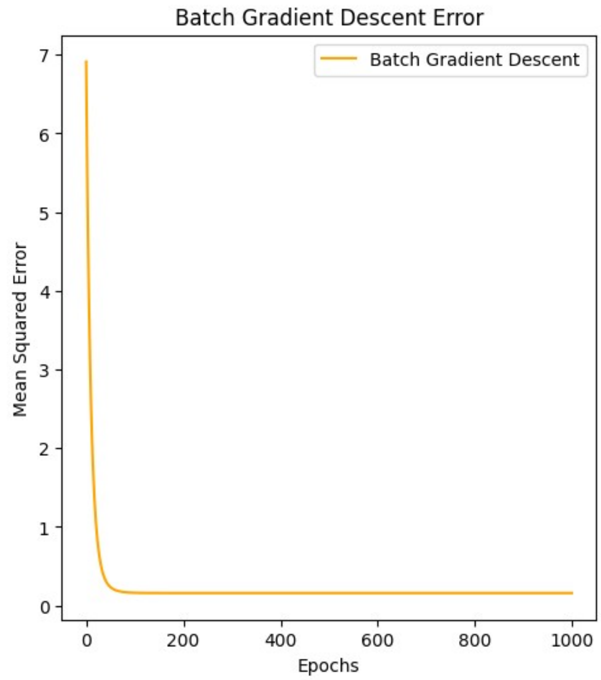
Lab1

code for implementing the Delta Learning Rule using both Stochastic Gradient Descent (SGD) and Batch Gradient Descent

```
import numpy as np
import matplotlib.pyplot as plt

X = np.array([0.5, 2.5])
Y = np.array([0.2, 0.9])
learning_rate = 0.01
epochs = 1000

np.random.seed(42)
weights = np.random.rand(2)
bias = np.random.rand()

def predict(X, weights, bias):
    return np.dot(X, weights) + bias

def calculate_error(Y, Y_pred):
    return np.mean((Y - Y_pred) ** 2)

def stochastic_gradient_descent(X, Y, weights, bias, learning_rate,
epochs):
    errors = []
    for epoch in range(epochs):
        for i in range(len(X)):
            y_pred = predict(X[i], weights, bias)
            error = Y[i] - y_pred
            weights += learning_rate * error * X[i]
            bias += learning_rate * error
        Y_pred = predict(X, weights, bias)
        mse = calculate_error(Y, Y_pred)
        errors.append(mse)
    return weights, bias, errors

def batch_gradient_descent(X, Y, weights, bias, learning_rate,
epochs):
    errors = []
```

```python
    for epoch in range(epochs):
        Y_pred = predict(X, weights, bias)
        error = Y - Y_pred
        weights += learning_rate * np.dot(X, error) / len(X)
        bias += learning_rate * np.mean(error)
        Y_pred = predict(X, weights, bias)
        mse = calculate_error(Y, Y_pred)
        errors.append(mse)
    return weights, bias, errors

weights_sgd, bias_sgd, errors_sgd = stochastic_gradient_descent(X, Y,
weights.copy(), bias, learning_rate, epochs)
weights_bgd, bias_bgd, errors_bgd = batch_gradient_descent(X, Y,
weights.copy(), bias, learning_rate, epochs)

plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)
plt.plot(errors_sgd, label='SGD')
plt.xlabel('Epochs')
plt.ylabel('Mean Squared Error')
plt.title('Stochastic Gradient Descent Error')
plt.legend()

plt.subplot(1, 2, 2)
plt.plot(errors_bgd, label='Batch Gradient Descent', color='orange')
plt.xlabel('Epochs')
plt.ylabel('Mean Squared Error')
plt.title('Batch Gradient Descent Error')
plt.legend()

plt.show()

print(f'Final weights (SGD): {weights_sgd}')
print(f'Final bias (SGD): {bias_sgd}')
print(f'Final weights (Batch GD): {weights_bgd}')
print(f'Final bias (Batch GD): {bias_bgd}')
```






```
Final weights (SGD): [0.34823306 0.3489669 ]
Final bias (SGD): [0.02858248 0.02709461]
Final weights (Batch GD): [0.31404721 0.89022139]
Final bias (Batch GD): -1.6416679993432692
```