ml-practical-4-piyusha

October 9, 2025

Piyusha Supe (BE-B-23CO315)

LP3_ML_Practical_4

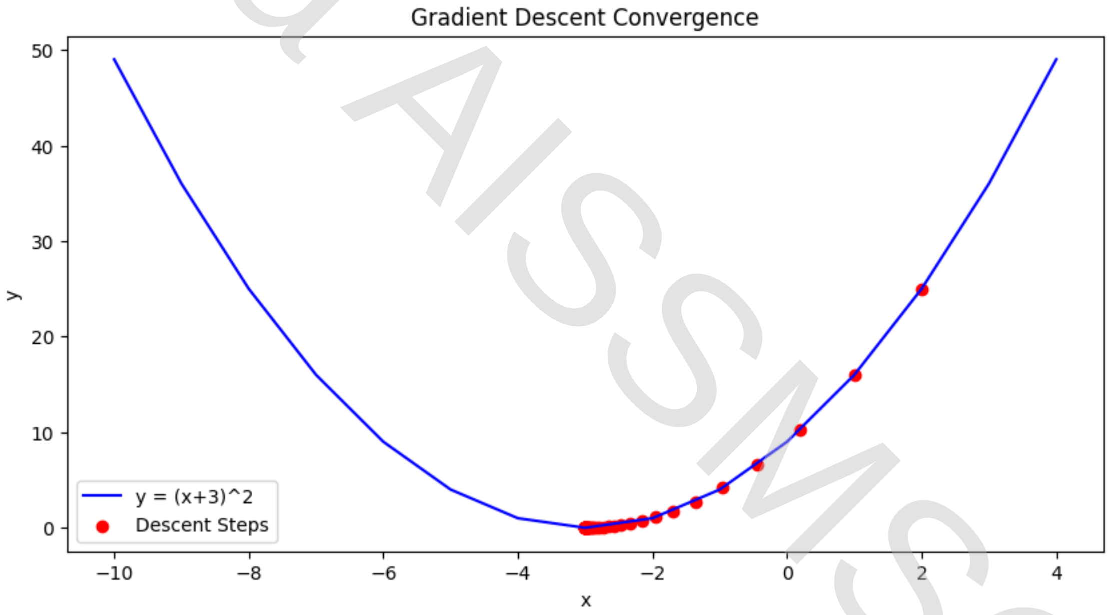
Implement Gradient Descent Algorithm to find the local minima of a function. For example, find the local minima of the function y=$(x+3)^2$ starting from the point x=2.

[1]:

```python
import matplotlib.pyplot as plt

# Function and its derivative
def f(x):
    return (x + 3)**2

def df(x):
    return 2 * (x + 3)

# Initialize parameters
x_old = 2        # Starting point
alpha = 0.1      # Learning rate
precision = 0.00001
max_iters = 100

# Lists for visualization
x_list = [x_old]
y_list = [f(x_old)]

# Gradient Descent Loop
for i in range(max_iters):
    grad = df(x_old)
    x_new = x_old - alpha * grad

    x_list.append(x_new)
    y_list.append(f(x_new))

    if abs(x_new - x_old) < precision:
        break
    x_old = x_new

# Display results
```

```python
print(f"Local minimum occurs at x = {x_new:.5f}")
print(f"Minimum value of function y = {f(x_new):.5f}")

# Visualization
plt.figure(figsize=(10,5))
x_vals = [i for i in range(-10, 5)]
y_vals = [f(i) for i in x_vals]
plt.plot(x_vals, y_vals, label='y = (x+3)^2', color='blue')
plt.scatter(x_list, y_list, color='red', label='Descent Steps')
plt.title('Gradient Descent Convergence')
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.show()
```

```
Local minimum occurs at x = -2.99996
Minimum value of function y = 0.00000
```



```
[ ]:
```