

Beispiel für Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent (SGD) mit Lernrate 0.01 und Schwellwert 0.0001

In diesem Beispiel verwenden wir das Stochastic Gradient Descent (SGD)-Verfahren, um für die Funktion $f(x) = x^4 - 4x^2 + x + 3$ das globale Minimum zu finden.

```
In [9]: # Importiere notwendige Libraries
import numpy as np
import matplotlib.pyplot as plt

# Funktion f(x) und ihre Ableitung f'(x)
def f(x):
    return x**4 - 4 * x**2 + x + 3

def f_prime(x):
    return 4 * x**3 - 8 * x + 1

#check versions of libraries
print('numpy version is: {}'.format(np.__version__))

# Import library time to check execution date+time
import time
print("*****current date and time *****")
print("Date an Time:",time.strftime("%d.%m.%Y %H:%M:%S"))
print("Erste Schritt sind Okay")

numpy version is: 1.18.1
*****current date and time *****
Date an Time: 17.09.2024 14:53:09
Erste Schritt sind Okay
```

```

In [13]: # Definiere SGD Funktion

# Stochastic Gradient Descent (SGD) Funktion
def stochastic_gradient_descent(start_x, learning_rate, epsilon):
    x = start_x
    iterations = 0
    change_in_f = float('inf')
    x_history = [x] # List to store the history of x values

    while change_in_f > epsilon:
        prev_f = f(x)
        x = x - learning_rate * f_prime(x)
        curr_f = f(x)
        change_in_f = abs(curr_f - prev_f)
        x_history.append(x) # Save the new x value after each update
        iterations += 1

    return x, curr_f, iterations, x_history

# Set initial conditions
start_x = 1.5
learning_rate = 0.01
epsilon = 0.00001

# Run SGD
result_x, result_f, total_iterations, x_history = stochastic_gradient_descent(
    start_x, learning_rate, epsilon)

# Print the results
print(f"Minimales x: {result_x}")
print(f"Wert der Funktion f(x) an diesem Punkt: {result_f}")
print(f"Anzahl der Iterationen: {total_iterations}")

# Plot the function and the SGD path
x_vals = np.linspace(-2, 2, 400)
y_vals = f(x_vals)

# Das Skript führt SGD durch, bis die Änderung in x kleiner als der Schwellwert ist,
# gibt den von x und die Anzahl der Iterationen und den Funktionswert an diesem Punkt an:

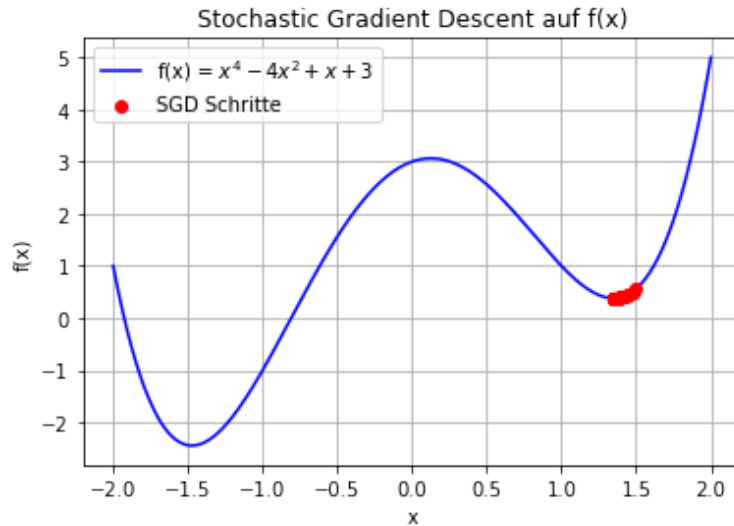
plt.plot(x_vals, y_vals, label="f(x) = $x^4 - 4x^2 + x + 3$", color='blue')
plt.scatter(x_history, [f(x) for x in x_history], color='red', zorder=5, label="SGD Schritte")
plt.plot(x_history, [f(x) for x in x_history], color='red', linestyle='--', alpha=0.6)
plt.title("Stochastic Gradient Descent auf f(x)")
plt.xlabel("x")
plt.ylabel("f(x)")
plt.legend()
plt.grid(True)
plt.show()

```

Minimales x: 1.3489888009680664

Wert der Funktion $f(x)$ an diesem Punkt: 0.381471370858105

Anzahl der Iterationen: 28



```
In [14]: # print current date and time
```

```
import time
print("*****current date and time *****")
print("Date an Time:",time.strftime("%d.%m.%Y %H:%M:%S"))
print ("Ende PGM ist Okay")
```

```
*****current date and time *****
```

```
Date an Time: 17.09.2024 14:56:19
```

```
Ende PGM ist Okay
```