1_gradient_descent

May 24, 2019

```
In [ ]: import numpy as np
        import matplotlib.pyplot as plt
```

```
In [ ]: x= np.arange(-10,10,1)
        f_x = x ** 2

        plt.plot(x, f_x)
        plt.show()
```

```
In [ ]: x_new = 10
        derivative = []
        y = []
        learng_rate= 0.1
        for i in range(100):
            old_value = x_new
            derivative.append(old_value - learng_rate * 2 * old_value)
            x_new = old_value - learng_rate *2* old_value
            y.append(x_new ** 2)
```

```
In [ ]: plt.plot(x, f_x)
        plt.scatter(derivative, y)
        plt.show()
```

```
In [ ]: def sin_function(x):
            return x * np.sin(x ** 2) + 1

        def derivitive_f(x):
            return np.sin(x**2) + 2 * (x **2) * np.cos(x ** 2)
```

```
In [ ]: x= np.arange(-3,3,0.001)
        f_x = sin_function(x)
```

```
In [ ]: plt.plot(x, f_x)
        plt.show()
```

```
In [ ]: derivitive_f(3)
In [ ]: x_new = 1
        derivative = []
        y = []
        learng_rate= 0.01
        for i in range(10000):
            old_value = x_new
            x_new = old_value - learng_rate * derivitive_f(old_value)
            derivative.append(x_new)
            y.append(sin_function(x_new))

        plt.plot(x, f_x)
        plt.scatter(derivative, y)
        plt.show()

In [ ]:
```