Notebook Assignment 1 - Due Date January 27:

Python (updated November 14, 2024)

Background: This problem asks you to implement a gradient descent algorithm in Python. Stochastic versions of this algorithm play an important rule in training deep neural networks. A gradient descent algorithm is used to search for the minimum of a differentiable function, $f : \mathbb{R}^n \to \mathbb{R}$. This is done by recursively computing an estimate of the minimum by taking the current estimate in the kth step, $x_k \in \mathbb{R}^n$ and computing a new estimate that moves along the negative direction of the gradient vector evaluate at x_k to obtain the $k+1$st estimate of the minimum. This recursive step may therefore be written as

$$x_{k+1} = x_k - \eta \frac{\partial f(x_k)}{\partial x} \tag{1}$$

The real parameter $\eta > 0$ is a hyperparameter called the *learning rate*. The algorithm starts with an initial guess, x_0, for the minimum (usually chosen at random) and then repeatedly executes equation (1) until a *termination condition* is satisfied. The termination condition is either 1) that the number of recursions (`iter`) exceeds a specified maximum number of iterations (`maxiter`) *or* that 2) the Euclidean 2-norm of the difference between successive estimates,

$$|x_{k+1} - x_k| \leq \texttt{tol}$$

is less than a specified *tolerance* limit, `tol`.

Problem Statement: Use the gradient descent recursion in equation (1) to search for the global minimum of the function

$$f(x) = x^4 - 5x^2 - 3x$$

Assume a starting estimate $x_0 = 0$. The termination conditions are $\texttt{maxiter} = 50$ and $\texttt{tol} = 10^{-5}$. Your notebook should address the following

(1) (4 points) Determine the gradient and Hessian of $f(x)$ and use those two things to 1) identify the extreme points of the function, 2) identify which extreme points are local minima, and 3) identify which extremal point is the global minimum.

(2) (4 points) Write a Python function

```
def gradient_descent(func, x0, learn_rate, maxiter, tol):
```

whose input arguments are

- `func` is the name of a function that takes input $x \in \mathbb{R}^n$ and returns $f(x) \in \mathbb{R}$ using equation (1).
- `x0` is a float point variable set to the initial estimate, $x_0 = 0$.
- `learn_rate` is the learning rate η in equation (1)
- `maxiter` and `tol` are the hyperparameters for the termination condition.

The function returns

- the gradient computed after the termination condition has been satisfied
- a numpy array, `history`, whose entries are the x_k computed in each recursion of the algorithm.

(3) (4 points) Use your function to plot the time history of the estimates as a function of iteration count for the following set of learning rates

$$\eta = 0.05, 0.10, 0.150.20, \text{ and } 0.25$$

On each plot print out in the title the ”final” estimate returned when the termination condition was satisfied. Describe the behavior that you see in all of these cases with respect to the gradient search’s ability to find the global minimum.