1 Exercise 1

Gradient descent is an iterative procedure to find the local minimum of a function $f(x)$ by successively looking at values of the function proportional to the direction of the gradient of the function. The iteration stops when the result from two successive steps yields a difference below a specified threshold. This means that once the local minimum is reached, movement along the gradient of a given function will not yield significantly different results. For a scalar function with vector argument inputs, this procedure can be written mathematically as:

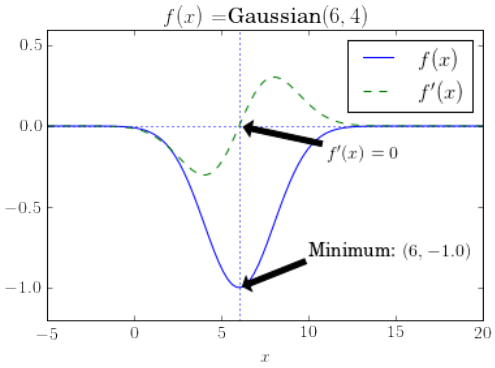
$$x_{n+1} = x_n - \eta * \nabla f(x_n).$$

A start value x_0 and a step size η is given to start the procedure, as well as a threshold β. When $|x_{n+1} - x_n| < \beta$, the procedure stops. It is possible to define different step sizes η for each iteration of the gradient descent procedure, but this analysis assumes a constant step size.

Testing this gradient descent optimizer on an upside-down Gaussian function with mean 6 and variance 4, we expect the minimum to occur at $(6, -1)$, since there is one global minimum. If the function and gradient are defined as:

$$f(x) = -e^{\frac{(x-6)^2}{8}} \qquad \nabla f = \frac{x-6}{4} e^{\frac{(x-6)^2}{8}}$$

(see Figure 1a), then the gradient descent algorithm always converges to the correct global minimum. For functions with one minimum, the larger the step value, the faster the algorithm will converge, regardless of how far the initial guess is from the minimum. The threshold dictates how close the numerical solution is from the analytical solution (see Figure 1b).



(a) Graph of $f(x)$ and its gradient

Guess	Step	Iterations	Minimum
5	0.5	21	(5.934, -.999)
5	1	13	(5.974, -1.000)
2	0.5	47	(5.938, -1.000)
2	5	4	(5.998, -1.000)

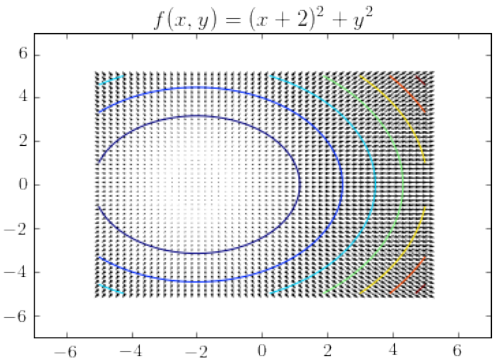
(b) Parameters and results of gradient descent. The threshold was always 0.01

Figure 1: Results for $f(x)$ =-Gaussian$(6, 4)$

The same algorithm can be applied to scalar functions with vector inputs. For example,

$$f(x, y) = (x + 2)^2 + y^2 \qquad \nabla f = 2(x + 2) + 2y$$

And has one global minimum. In choosing parameters, the smaller the step size and the smaller the threshold, the more iterations are required for the algorithm to converge.



(a) $f(x, y)$, ∇f, and the estimated minimum

Guess	Step	Thresh	Iterations
(0, 0)	0.5	0.01	2
(5, 5)	0.5	0.01	2
(5, 5)	0.1	0.01	25
(5, 5)	0.1	0.001	35

(b) Parameters and results of gradient descent. The minimum always came out to be $(-2, 0, 0)$

Figure 2: Using gradient descent to estimate the minimum of a scalar function with vector inputs

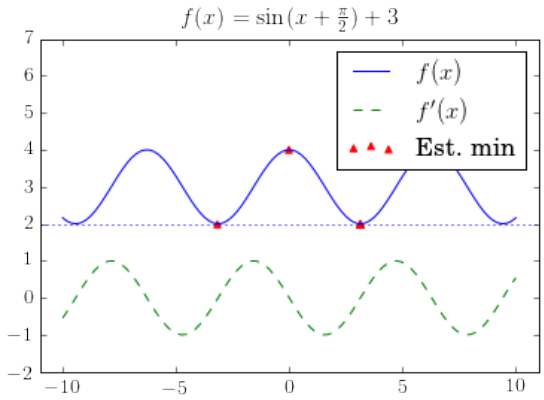
For functions without a global minimum (non-convex functions), the choice of input parameters dictates not only how fast the algorithm converges, but also the value at which it converges. For example, the function

$$f(x) = \sin(x + \frac{\pi}{2}) + 3 \qquad \nabla f = \cos(x + \frac{\pi}{2})$$

has local minima at $(2(n + 1)\pi, 2)$ for all n (see Figure 2a). When the initial guess for gradient descent is chosen to be closer to the local minima at $-\pi$, the algorithm estimates that to be the local minimum, and vice versa for π. If the initial guess is given at 0, which is a local maximum, the algorithm does not descend and stops after one iteration.

The gradient of a function can either be specified analytically or numerically approximated with central differences using $\delta_h[f](x) = f(x + 0.5h) - f(x - 0.5h)$. From a numerical standpoint, using either the analytical or numerical approximation to the gradient is approximately equivalent, given the correct choice of step size (see Figure 4).

Software packages like SciPy implement a different algorithm for estimating minima. Table 1 gives a comparison of the number of iterations of the algorithm needed to find a solution. In general, the BFGS optimizer in SciPy performs better than the simple gradient descent algorithm presented here. Both algorithms came up to the same solution within a few significant digits.

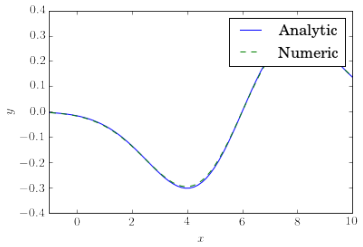


(a) A sinusoid and it's derivative, together with the estimated minima

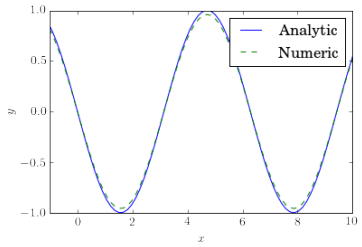
Guess	Step	Thresh	Iterations	Minimum
5	0.5	0.01	9	$(\pi, 2)$
-5	0.5	0.01	9	$(-\pi, 2)$
0	0.5	0.01	1	$(0, 4)$
0.5	0.5	0.01	12	$(\pi, 2)$
0.2	0.5	0.01	14	$(\pi, 2)$
0.7	0.5	0.01	11	$(\pi, 2)$

(b) Parameters and results of gradient descent

Figure 3: Using gradient descent on a function with many local minima



(a)



(b)

(x, y)	Analytic ∇f	Numeric
(-2, -2)	(0, -4)	(0, -4)
(-2, -1)	(0, -2)	(0, -2)
(-2, 0)	(0, 0)	(0, 0)
(-2, 1)	(0, 2)	(0, 2)
(-2, 2)	(0, 4)	(0, 4)

(c)

Figure 4: The step size for calculating the central difference was 1. (a) The analytic and numeric approximations of Figure 1a. (b) The analytic and numeric approximations of Figure 2a. (c) The analytic and numeric approximations of Figure 3a.

Function	Algorithm	Guess	Iterations
$f(x) = -\text{Gaussian}(6, 4)$	GD	5	21
$f(x) = -\text{Gaussian}(6, 4)$	BFGS	5	6
$f(x) = \sin(x + \frac{\pi}{2}) + 3$	GD	5	9
$f(x) = \sin(x + \frac{\pi}{2}) + 3$	BFGS	5	4
$f(x, y) = (x + 2)^2 + y^2$	GD	(5, 5)	25
$f(x, y) = (x + 2)^2 + y^2$	BFGS	(5, 5)	2

Table 1: Steps to convergence. GD is gradient descent, BFGS is from SciPy.

2 Exercise 2

This is a test.

3 Exercise 3

This is a test.

4 Exercise 4

This is a test.