CSE 4345 – Homework #7

Assigned:	Thursday, April 4, 2019
Due:	Thursday, April 11, 2019 at the end of class

Note the following about the homework:

1. You must show your work to receive credit.
2. If your submission has more than one page, staple the pages. If I have to staple it, the cost is 10 points.

Assignment:

1. (hand solution) Minimize $x^3 - 5x^2 - 10x + 5$ using Newton's method (1D optimization version) with

   (a) $x_0 = 0$. Stop after finding x_4; then find $f(x_4)$.

   (b) $x_0 = 2$. Stop after finding x_4; then find $f(x_4)$.

   Which, if either, of these initial values results in a minimum? Remember, you can use the second derivative to determine this.

2. (hand solution) Minimize
$$f(x_1, x_2) = x_1^4 - 2x_1^2 x_2 + x_2^2$$
   using gradient descent with $\vec{x}_0 = (1, 2)$. Use $\alpha = 0.1$ and stop after producing $\vec{x}_3$. What is $f(\vec{x}_3)$?

3. Use Newton's method (multidimensional optimization version) to find critical points of
$$f(x_1, x_2) = x_1^3 - x_1 x_2 + x_2^3$$
   when

   (a) $\vec{x}_0 = (1, 1)$. Stop after finding $\vec{x}_4$; then find $f(\vec{x}_4)$.

   (b) $\vec{x}_0 = (-2, -1)$. Stop after finding $\vec{x}_4$; then find $f(\vec{x}_4)$.

   Be clear what the general form of the gradient and Hessian are for this problem. Then for each

   (a) Find the eigenvalues of the Hessian matrix at the critical point.

   (b) Determine if the critical point is a local minimum, local maximum, or saddle point.

   (c) What is the definiteness of the Hessian at this point?

This would be painful to do strictly by hand, so you can use Numpy to assist with the math. Don't use any Numpy function that basically just solves an optimization problems for you, but you can write a program to carry out the steps of the algorithm.

Show enough of the steps that someone (e.g., your future self) looking at your work could understand the process. Remember that you will have to do this by hand on the exam, so do the work and transcribe the process in such a way that you are preparing for that.

4. **(application: artificial intelligence)** This problem is based on the example in [RN10, p. 131] with some simplifications.

   Suppose we need to build a hospital to be shared by several cities. We don't want it to be too far away from any particular city. The question is where to place the hospital. One way to decide is to place it such that we minimize the sum of the squared distances (the 2-norm) from the hospital to the cities.

   (a) Write a Python program to use gradient descent to choose the placement of the hospital.

   (b) On the course website is the skeleton file to use, `gradient.py`. The main section of the file contains

      i. a 2D Numpy array of the coordinates (as x,y pairs). This array could have more or fewer values.

      ii. a 1D Numpy array of the starting coordinates for placing the hospital. There will always be a single hospital to place.

      iii. the learning rate, `alpha`

      iv. the tolerance, `tol`, to use for deciding how long to iterate. You will iterate as long as

      $$||x_{k+1} - x_k||_2 > \text{tol}$$

      These values are passed to the function `positionHospital()`, which you will write. If I were to change any of the values for the variables described above, your program should work correctly.

      The function should return a 1D Numpy array of the positions of the hospital from the initial position to the final position.

   (c) On each iteration, print

      i. the iteration count

      ii. the value of the function with the current hospital coordinates

      iii. the value of the gradient with the current hospital coordinates

      iv. the coordinates of the hospital

      An example is

```
k =  13, fVal = 17362.59, grad = [94.71, 27.06]  h = [24.11, 61.89]
```

   (d) You should also write `plotCoor()`, which displays the locations of the cities, as circles, and all positions of the hospital as an 'x'. Figure 1 shows an example.

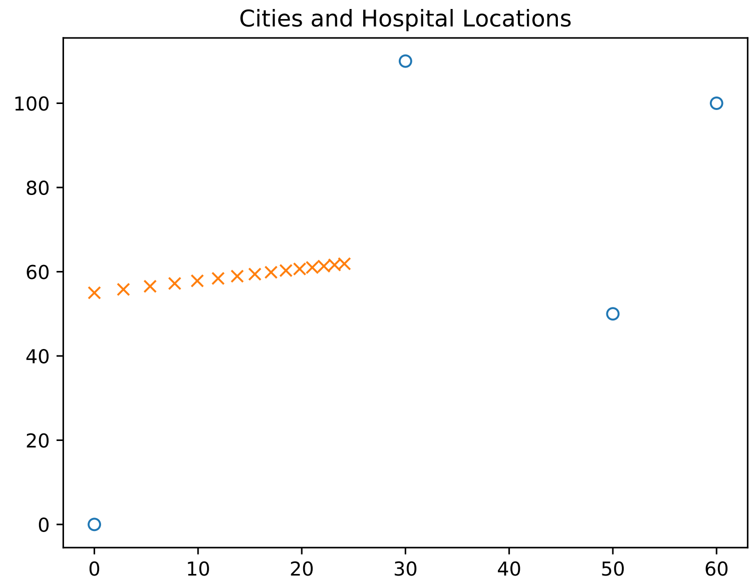

Figure 1: Using gradient descent to place hospital

(e) While you should solve this problem using gradient descent, you can check your work with the knowledge that when there is only one hospital to be placed then the solution is the mean of the coordinates of the cities. Note, however, that depending on your stopping criteria the solution found using a numerical method may not match the mean exactly.

General requirements about the Python problems:

a) As a comment in your source code, include your name.

b) The Python program should do the work. Don't perform the calculations and then hard-code the values in the code or look at the data and hard-code to this data unless instructed to do so.

c) The program should not prompt the user for values, read from files unless instructed to do so, or print things not specified to be printed in the requirements.

To submit the Python portion, do the following:

a) Create a directory using your net ID in lowercase characters plus the specific homework. For example, if your net ID is `abc1234` and the homework is `hw04`, then the directory should be named `abc1234-hw04` (zero-pad the number if necessary to make it two digits).

b) Place your .py files in this directory.

c) Do not submit the data files unless instructed to do so.

d) Zip the directory, not just the files within the directory. You must use the zip format and the name of the file (using the example above) will be `abc1234-hw04.zip`.

e) Upload the zip'd file to Blackboard.

References

[RN10] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Upper Saddle River, New Jersey, 3^{rd} edition, 2010.