TRINITY COLLEGE DUBLIN

School of Computer Science and Statistics

Week 2 Assignment CS7CS2 Optimisation for Machine Learning

Rules of the game:

- **Its ok to discuss with others, but do not show any code you write to others. You must write answers in your own words and write code entirely yourself. All submissions will be checked for plagiarism.**
- **Reports must be typed (no handwritten answers please) and submitted as a separate pdf on Blackboard (not as part of a zip file please).**
- **Important: For each problem, your primary aim is to articulate that you understand what you're doing - not just running a program and quoting numbers it outputs. Long rambling answers and "brain dumps" are not the way to achieve this. If you write code to carry out a calculation you need to discuss/explain what that code does, and if you present numerical results you need to discuss their interpretation. Generally most of the credit is given for the explanation/analysis as opposed to the code/numerical answer. Saying "see code" is not good enough, even if code contains comments. Similarly, standalone numbers or plots without further comment is not good enough.**
- **When your answer includes a plot be sure to (i) label the axes, (ii) make sure all the text (including axes labels/ticks) is large enough to be clearly legible and (iii) explain in text what the plot shows.**
- **Include the source of code written for the assignment as an appendix in your submitted pdf report. Also include a separate zip file containing the executable code and any data files needed. Programs should be running code written in Python i.e. so that we can unzip your submission and just directly run it to check that it works. Keep code brief and clean with meaningful variable names etc.**
- **Reports should typically be not more than aboput 5 pages, with 10 pages the absolute upper limit (excluding appendix with code). If you go over 10 pages then the extra pages will not be marked.**

Assignment

(a) Begin by looking at the function $y(x) = x^4$.

  (i) Using sympy obtain an expression for the derivative dy/dx. Show all the steps that you used to obtain the expression.

  (ii) For a range of values of x calculate the derivative value using the expression from (i). Also calculate the derivative value estimated using a finite difference with perturbation $\delta = 0.01$ on x. Plot these values on the same plot and compare. Show the code that you used to calculate the finite difference estimate.

  (iii) Now vary the size of the perturbation δ of x in the finite difference approximation from 0.001 to 1, and plot the difference between the derivative estimate and the exact derivative values caclulated using the sympy expression. Use a single plot here so that the effect of changing δ is easier to compare. How does changing δ affect the accuracy of the estimated derivative? Why?

(b) (i) Write a short python programme that implements gradient descent with a fixed step size α. Explain your code.

  (ii) With initial value $x = 1$ and step sizes $\alpha = 0.1$ run your gradient descent programme with the function $y = x^4$. Plot how x and $y(x)$ vary with each gradient descent iteration. Discuss.

  (iii) Now repeat (ii) for a range of initial values for x and step sizes α. Be sure to choose a large enough range of step sizes that you see a variety of behaviours (including non-convergence). How does the choice of x and α values affect the convergence time and accuracy? Why?

(c) (i) Change to using the function $y(x) = \gamma x^2$ where γ is a parameter that you choose. Holding the step size α constant, how does changing γ affect the convergence of your gradient descent programme? Why?

(ii) Repeat for the function $y(x) = \gamma|x|$.