TRINITY COLLEGE DUBLIN

School of Computer Science and Statistics

Week 4 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.**

Obtaining Functions

- Download the functions to use in the assignment from `https://www.scss.tcd.ie/Doug.Leith/CS7DS2/week4.php`. Important: You must fetch your own copy of the functions, do not use the dataset downloaded by someone else.
- Please cut and paste the functions and include in your submission.
- Use sympy to obtain an expression for the derivative of each function, and also include this in your submission. Note that the derivative of Max(0,x) is the Heaviside function

$$\theta(x) = \begin{cases} 0 & x < 0 \\ 1 & x > 0 \end{cases}$$

.

.

Assignment

(a) Your first task is to implement the following gradient descent updates that we studied in the lectures:

(i) Polyak step size

(ii) RMSProp

(iii) Heavy Ball

(iv) Adam

Give code snippets for each algorithm (to save space don't include boilerplate code, just the part that calculates the update step to take - but do include the full code in an appendix). Explain how each code snippet implements the corresponding algorithm.

(b) Now apply these algorithms to each of the functions that you downloaded to show the impact of changing the following parameters, and discuss:

(i) α and β in RMSProp.

(ii) α and β in Heavy Ball.

(iii) α, β_1 and β_2 in Adam.

Try e.g. $\beta = 0.25$ and 0.9 and a wide range of α values e.g. 0.01 up to a value that causes each algorithm to stop converging. To save space plot multiple curves of function value vs iteration number on a single plot. Where appropriate, to support your discussion use plots of e.g. the step size vs iteration number, contour plots showing how the function value $f(x, y)$ changes with each iteration.

(c) Now you'll look at the ReLu function $Max(0, x)$.

(i) For initial condition $x = -1$ run each of the algorithms you've implemented (use reasonable parameter values for each algorithm based on part (b)). What happens, and why?

(ii) Now look at initial condition $x = +1$ and repeat. How does the behaviour change, and why?

(iii) What happens when you change the initial condition to be $x = +100$?