TRINITY COLLEGE DUBLIN

School of Computer Science and Statistics

Week 8 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 ML Data and Model

- Download the python code at `https://www.scss.tcd.ie/Doug.Leith/CSU44061/week8.py` that you already used in the week 8 Machine Learning assignment. This uses keras to load the CIFAR10 dataset (50K small 32×32 colour images plus labels), it then builds a convolutional network, trains it and evaluates the confusion matrix of its predictions. Modify the code to calculate a suitable cost function over the test data e.g. the logistic (or cross-entropy) loss or AUC.

.

Assignment

(a) (i) Using python implement the global random search algorithm. This takes as input the number of parameters n, the min and max value of each parameter (for the i'th parameter call these l_i and u_i), and the number N of samples to take. To generate a sample the search algo draws the i'th parameter value uniformly at random between l_i and u_i, by doing this for all n parameter values a full parameter vector is generated. The cost function is evaluated at this vector, if the cost function is the lowest seen so far then that is recorded and this process is repeated N times.

(ii) Apply your random search algorithm to the two functions that you looked at in week 4 (give these function again here). How does its performance compare with gradient descent? Note that for a fair comparison you'll need to compare the cost function value vs the running time of each algo (although this might be hard to measure accurately, why?) and the number of function and derivative evaluations. Also, due to the randomness in the search you'll need to look at the performance across multiple runs e.g. plot the average and std dev of the cost function vs iterations, time etc rather just plotting a single run.

(b) (i) Now modify your global random search algorithm so that after calculating N parameter vectors and calculating the cost functions it keeps the M best parameter vectors out of these (i.e. the ones with the lowest cost function value). It then generates N random parameter vectors in a neighbourhood about each of these M points, keeps the M best of these and repeats. This is similar to the population-based sampling approach that we discussed in the lectures.

(ii) Apply this new random search algorithm to the two functions that you looked at in week 4, and compare the performance to that of random search and gradient descent. Using contour plots of the cost functions superimpose the choices of parameter values vs time made by these algorithms and compare.

(c) Now apply your two random search algorithms to choose hyperparameters for the conv net model that you downloaded. As hyperparameters use (i) the mini-batch size, (ii) adam parameters α, β_1, β_2 and (iii) number of epochs. Compare their performance.