Assignment 4

22 November, 2022

Due date: 2 December

MSAF accommodations 5 December
SAS accommodations 7 December, 12:30pm

Instructions

- Submit to Avenue a **PDF file** containing your solutions.
  Do not submit files in other formats e.g. IMG, PNG.
- Name your MATLAB files **exactly** as specified.
- Name your PDF file **Lastname-Firstname-studentnumber.pdf**.
- Do not submit zipped files.

Problem 1 [15 points]

(a) [5 points] Modify the file `netbpfull.m` such that it classifies points as described below.

- The points are stored in a file `dataset.mat`.

  Load this file as `load('dataset.mat');`

  It contains an array `X` of data points and an array `Y` with their labels.

- Modify the function `cost` such that it classifies the output of the network and also returns

  $$\text{accuracy} = \frac{\text{number of points classified correctly}}{\text{total number of points}}$$

  The signature of this function becomes

  `function [costval,accuracy] = cost(W2,W3,W4,b2,b3,b4)`

- `netbpfull.m` should plot accuracy versus number of iterations, in addition to cost versus number of iterations. That is, it should produce four plots in total.

- The training should stop if accuracy of 0.97 is reached; otherwise it should continue to `Niter`.

(b) [5 points] Select number of neurons in layers 2 and 3, learning rate, and number of iterations such that when the execution is finished, the accuracy on these training data is ≥ 0.97.

(c) [5 points] Use the same parameters as in item b, but now increase the batch size to 4 (currently it is of size one). Does the training time improve with a batch size of 4?

Submit

- PDF: `netbpfull.m` after you have done item c and highlight the parts you have changed.

  The plots from items b and c, except the Figure 1 showing the data.

  Report the parameters in item b that you have used, and the achieved accuracies in items b and c.

  If you reach 0.97, report also the number of iterations when 0.97 is reached.

- Avenue: your modifed `netbpfull.m`.

Bonus [10 points] Modify `nlsrun.m` such that it runs on the points from `dataset.mat`. Submit the produced plot and the modified code of `nlsrun.m` in the PDF and and also submit this code to Avenue.

Problem 2 [5 points] Implement in Matlab the bisection method for finding a root of a scalar equation. Consider the polynomial

$$\begin{aligned} f(x) &= (x-2)^9 \\ &= x^9 - 18x^8 + 144x^7 - 672x^6 + 2016x^5 - 4032x^4 + 5376x^3 - 4608x^2 + 2304x - 512. \end{aligned}$$

(a) [3 points] Write a Matlab script to evaluate this function at 100 equidistant points in the interval $[1.94, 2.07]$ using two methods:

- evaluate $(x-2)^9$ directly
- evaluate $x^9 - 18x^8 + 144x^7 - 672x^6 + 2016x^5 - 4032x^4 + 5376x^3 - 4608x^2 + 2304x - 512$ using Horner's method

Plot the results in two different plots. Explain the differences between the two plots.

(b) [2 points] Apply your bisection method to find a root starting with $[1.94, 2.07]$, tolerance 10^{-6}, and using the Horner's evaluation.

Obviously, $r = 2$ is a root of multiplicity 9. Can you determine a value for r such that $|r - 2| < 10^{-6}$?

Discuss your observations.

Submit

- PDF: your plots and discussion
- Avenue: all Matlab code. Name the main program `p2_main.m`. It should produce all required results.

Problem 3 [8 points] Implement Newton's method for systems of equations.

Each of the following systems of nonlinear equations may present some difficulty in computing a solution. Use Matlab's `fsolve` and your own implementation of Newton's method to solve each of the systems from the given starting point.

In some cases, the nonlinear solver may fail to converge or may converge to a point other than a solution. When this happens, try to explain the reason for the observed behavior.

Report (in your PDF) for `fsolve` and your implementation of Newton's method and each of the systems below, the number of iterations needed to achieve accuracy of 10^{-6} (if achieved).

(a)

$$\begin{aligned} x_1 + x_2(x_2(5 - x_2) - 2) &= 13 \\ x_1 + x_2(x_2(1 + x_2) - 14) &= 29 \end{aligned}$$

starting from $x_1 = 15$, $x_2 = -2$.

(b)

$$\begin{aligned} x_1^2 + x_2^2 + x_3^2 &= 5 \\ x_1 + x_2 &= 1 \\ x_1 + x_3 &= 3 \end{aligned}$$

starting from $x_1 = (1+\sqrt{3})/2$, $x_2 = (1-\sqrt{3})/2$, $x_3 = \sqrt{3}$.

(c)

$$\begin{aligned} x_1 + 10x_2 &= 0 \\ \sqrt{5}(x_3 - x_4) &= 0 \\ (x_2 - x_3)^2 &= 0 \\ \sqrt{10}(x_1 - x_4)^2 &= 0 \end{aligned}$$

starting from $x_1 = 1$, $x_2 = 2$, $x_3 = 1$, $x_4 = 1$.

(d)

$$\begin{aligned} x_1 &= 0 \\ 10x_1/(x_1 + 0.1) + 2x_2^2 &= 0 \end{aligned}$$

starting from $x_1 = 1.8$, $x_2 = 0$.

Problem 4 [3 points] Consider two bodies of masses $\mu = 0.012277471$ and $\widehat{\mu} = 1 - \mu$ (Earth and Sun) in a planar motion, and a third body of negligible mass (moon) moving in the same plane. The motion is given by

$$\begin{aligned} u_1'' &= u_1 + 2u_2' - \widehat{\mu}\frac{u_1 + \mu}{((u_1+\mu)^2 + u_2^2)^{3/2}} - \mu\frac{(u_1 - \widehat{\mu})}{\big((u_1 - \widehat{\mu})^2 + u_2^2\big)^{3/2}} \\ u_2'' &= u_2 - 2u_1' - \widehat{\mu}\frac{u_2}{((u_1+\mu)^2 + u_2^2)^{3/2}} - \mu\frac{u_2}{\big((u_1 - \widehat{\mu})^2 + u_2^2\big)^{3/2}}. \end{aligned}$$

The initial values are

$$\begin{aligned} u_1(0) &= 0.994, & u_1'(0) &= 0, \\ u_2(0) &= 0, & u_2'(0) &= -2.00158510637908252240537862224. \end{aligned}$$

Implement the classical Runge-Kutta method of order 4 and integrate this problem on [0,17.1] with uniform stepsize using 100, 1000, 10,000, and 20,000 steps. Plot the orbits for each case. How many uniform steps are needed before the orbit appears to be qualitatively correct?

Submit

- PDF: plots and discussion.

Problem 5 [5 points] The initial value problem

$$y' = \begin{pmatrix} -0.1 & -199.9 \\ 0 & -200 \end{pmatrix} y$$
$$t \in [0, 100], \quad y(0) = (2, 1)^T$$

has a solution

$$y_1(t) = e^{-0.1t} + e^{-200t}, \quad y_2(t) = e^{-200t}.$$

Integrate this problem with Matlab's `ode45` using absolute error tolerances 10^{-1}, 10^{-2},..., 10^{-12}. For each tolerance, show in a table the number of function evaluations, the error at $t = 100$, and the average stepsize over the interval of integration. Also, plot the stepsize versus t for tolerance 10^{-7}. Explain the stepsize behavior.

Integrate this problem with `ode23s`. Is this solver more efficient (on this problem) than `ode45`. Why?

Submit

- PDF: plot, table, and discussion