TC3020 Machine Learning
Assignment 4: Artificial Neural Networks (backprop)

Dr. Rafael Pérez Torres

October 1st, 2021

Abstract

In this assignment, you will implement the backpropagation learning step of an Artificial Neural Network.

1 Description

Overall goal: Implement the backpropagation step of the ANN classifier.

You must not employ specialized libraries (`sklearn` or similar) to produce your solution; it is ok to employ them to compare though.

You have access to a partially implemented source code for an ANN with support for multiple layers and different number of neurons in each of them. Take your time to study the code and understand its structure. There are four main elements:

- A launcher (`launcher.py`) that creates some instances of the ANN and runs them.
- An ANN class (`ann.py`) that implements the core aspects of an ANN. **You will be working in this file**.
- A utils script (`utils.py`) that defines some functions to read the dataset, plot results, etc.

2 Implementation

- Locate the TODO comments in the `ann.py` file and implement the required functionality.
- Pay attention to the dimensions of the parameters employed in the functions, so that you can be sure your implementation is returning arrays/values with the proper dimensions..
- **Debug as much as you can** to understand the code flow and identify issues.

Without modifying the parameters and configuration in launcher, you will get the following output for the provided datasets:

2.1 Example dataset

This is the one seen in the numerical example.

```
[[0.05 0.1 ]] predicted as [[0.75136507 0.77292847]]
final theta is [
    array([
    [0.35, 0.14886582, 0.19773165],
    [0.35, 0.24871137, 0.29742273]
    ]),
    array([
        [0.6, 0.18008518, 0.22874539],
        [0.6, 0.56439101, 0.6147833 ]
```

```
    ])
]
final cost is 1.6505285318122682
[[0.05 0.1 ]] predicted as [[0.69916895 0.78612171]]
```

See how we get the same values as in that example execution.

2.2 XOR dataset

```
final theta is [array([[ 0.5488135 ,  0.95417338,  0.90204995],
    [ 0.54488318, -3.77189155,  6.66244383],
    [ 0.43758721,  3.52770412,  3.55924584],
    [ 0.38344152,  3.76872723,  3.78987618],
    [ 0.56804456,  6.68319135, -3.83534674],
    [ 0.0871293 ,  3.07271269,  3.15387758]]),
    array([[ 0.77815675,  0.87161437,  1.14847215,  0.71068446,  0.34632749,
      0.92935881,  0.02192366],
    [ 0.63992102, -0.54331313,  2.94756796, -1.65199755, -1.78861651,
      2.55361862, -0.91796704],
    [ 0.45615033, -0.21812464, -5.84862528,  2.91780182,  3.19553836,
     -5.89933913,  2.32134459],
    [ 0.6818203 ,  0.23894948,  0.67283659,  0.32139552, -0.32999526,
      1.04524572,  0.36774798],
    [ 0.21038256, -0.4573598 , -5.72943848,  2.92181413,  3.40672112,
     -5.78950609,  2.43808733],
    [ 0.10204481, -0.13595683,  4.98111203, -2.55518598, -3.14998893,
      5.10225167, -1.81037305]]),
      array([[ 0.15896958,  1.24102332,  3.72131036, -7.97903954,  1.02651628,
     -7.87549383,  7.56505366],
    [ 0.09710128, -1.40063648, -3.9881624 ,  7.98016995, -1.43146536,
      8.17250572, -7.06451715]])]
final cost is 5.209273914993036e-05
[[0 0]] predicted as [[9.99970076e-01 3.00345411e-05]]
[[0 1]] predicted as [[2.46142976e-05 9.99975417e-01]]
[[1 0]] predicted as [[2.37694509e-05 9.99976469e-01]]
[[1 1]] predicted as [[9.99982446e-01 1.72984178e-05]]
```

2.3 Blobs dataset

```
final theta is [
  array([[ 0.22199317,  0.94063754, -0.74595822],
  [ 0.91861091,  1.97478159,  1.63979535]]),
  array([[ 0.76590786,  2.56948942,  1.89895978],
  [ 0.18772123, -4.23513606, -0.09555791],
  [ 0.44130922,  2.73780475,  2.82513439],
  [ 0.27408646,  1.96446058, -4.57666838],
  [ 0.62878791, -1.94365536,  1.29294904]]),
  array([[ 0.26581912,  0.58083945, -7.04955206, -0.02688059,  3.4759438 ,
   -2.7340618 ],
  [ 0.96393053, -0.43879987,  2.02259435,  0.60847564, -8.55385198,
    0.52938218],
  [ 0.77951459, -2.97430813,  5.00570555, -4.61836339,  3.93442076,
    1.19955659]])]
final cost is 0.18772726663395342
[[ 1 -9]] predicted as [[0.96236329 0.00156561 0.06717929]]
```

```
[[-4.   7.8]] predicted as [[0.00526761 0.92712694 0.06120099]]
[[-9.   4.5]] predicted as [[0.04664401 0.08527041 0.8445389 ]]
```

2.4 Moons dataset

```
final theta is [array([[ 0.5488135 ,  0.79151867, -0.57466833],
    [ 0.54488318, -1.55585079,  0.5369472 ],
    [ 0.43758721,  2.60289339,  2.7756662 ],
    [ 0.38344152,  3.20682205, -0.20798928]]),
    array([[ 0.56804456,  5.62896905, -7.85449412,  1.23065426,  4.61364348],
    [ 0.83261985, -6.31030556,  8.87614035,  1.70556553,  1.81953249],
    [ 0.46147936, 11.87234075,  5.30657111, -7.3647204 , -5.64364596],
    [ 0.94466892, -4.12816734,  3.20954153,  0.84208299,  1.85776155]]),
    array([[ 0.45615033,  7.36145501, -9.4333498 , 12.46040975, -6.53028663],
    [ 0.616934  ,  0.22706139,  1.90456492,  0.03906128,  1.38716852],
    [ 0.6976312 , -5.34512446,  5.88738276, -7.94027209,  3.85667945],
    [ 0.1289263 ,  3.96838899, -4.72179737,  5.93458639, -3.4596197 ]]),
    array([[  0.98837384, -14.93913179,   3.81295434,   9.58245705,
      -6.23597597],
    [  0.2532916 ,  14.61347316,  -4.49206871, -10.21021092,
       6.02413267]])]
final cost is 9.030247144403171e-05
[[-0.5  0.5]] predicted as [[9.99733048e-01 2.57611475e-04]]
[[1.  0.5]] predicted as [[9.99997733e-01 2.18278474e-06]]
[[0 0]] predicted as [[3.43569392e-07 9.99999668e-01]]
[[ 1.5 -0.5]] predicted as [[9.35398518e-07 9.99999108e-01]]
```

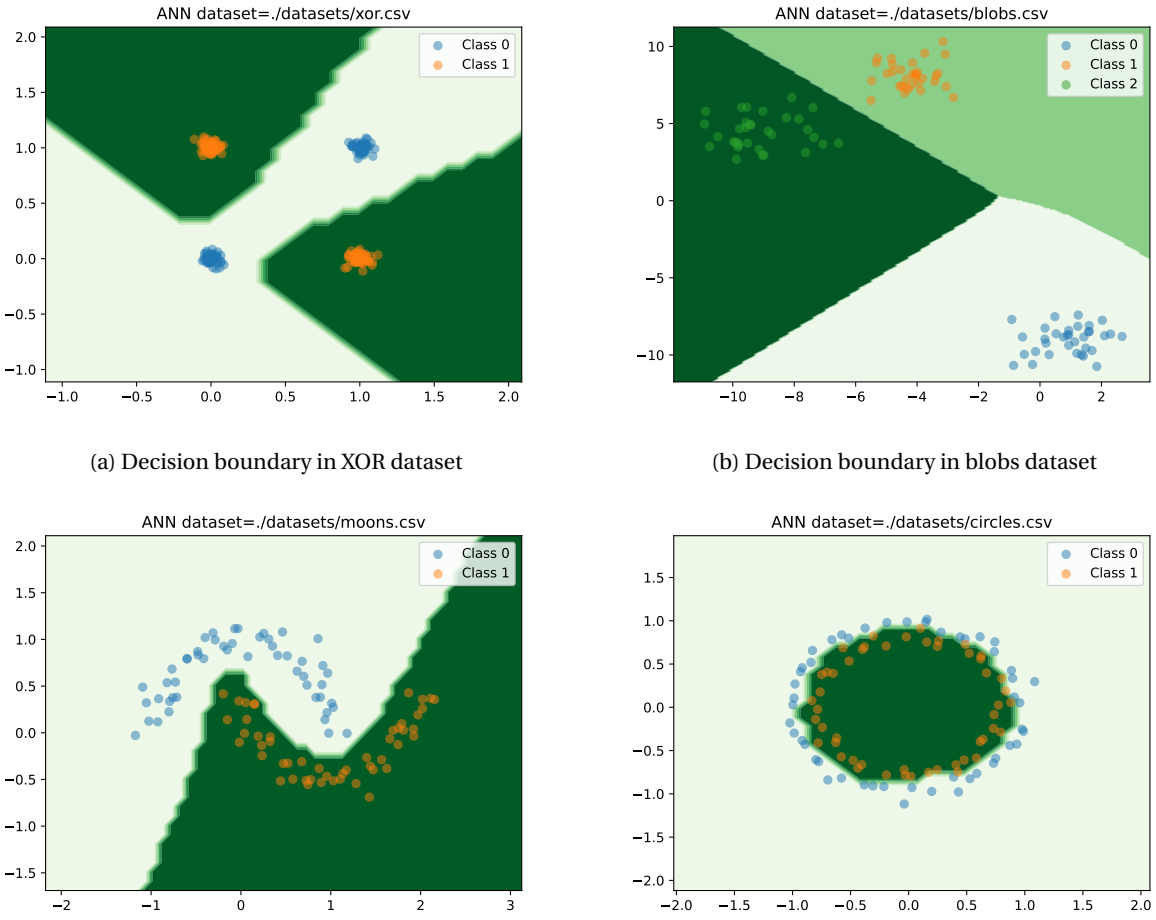
2.5 Circles dataset

```
final theta is [array([[ 0.5488135 ,  6.28232469, -4.65440391],
    [ 0.54488318,  1.991373  ,  1.58857333],
    [ 0.43758721,  1.32886533, -1.74806586],
    [ 0.38344152, -1.4617028 , -1.41775511],
    [ 0.56804456,  1.89719733, -2.08220106],
    [ 0.0871293 ,  4.36905716,  4.83879906],
    [ 0.77815675,  3.144532  , -1.57420873],
    [ 0.79915856, -2.22247344,  1.21573255],
    [ 0.11827443, -0.92672181, -1.38608565]]),
    array([[  0.94466892,  -0.52576898,  -4.32708157,   1.21790019,
       3.67511242,  -0.43337036,   5.03462721,  -4.83527299,
     -11.7393481 ,   8.18349123],
    [  0.616934  ,   0.73150136,   0.17064098,  -0.0773539 ,
       0.30877345,   0.16539605,   1.16884884,  -0.09883481,
       0.84354292,   0.30360291],
    [  0.1289263 ,   3.72622   ,   2.02375815,   4.85066324,
       2.15183732,   5.26526579,   0.04189823,   4.99125117,
      -6.24579416,   1.51722886],
    [  0.2532916 ,   1.28550345,   0.05377598,   0.61983636,
       0.10130918,   1.22639085,   0.38057328,   0.57803442,
      -0.49508711,   0.73138774],
    [  0.09710128,   4.66765354,   1.77849277,   5.60743592,
       2.91862981,   5.60655253,   0.40611326,   5.56964064,
      -6.61543799,   1.04359495],
    [  0.12019656,   0.10295602,  -3.71076541,   0.36969838,
```

```
       5.33094065,  -2.10633131,   3.24491275,  -4.25597426,
      -9.0109325 ,   7.85175599],
    [  0.09394051,  -1.98288495,   3.8619571 ,  -0.01374276,
      -5.25090077,  -1.16622041,   2.71797356,   0.43065454,
      -4.87319628,  -3.32261025],
    [  0.02010755,  -1.79526686,   2.59021913,  -0.07079111,
      -4.12687487,  -0.90239563,   3.84088426,  -0.6421551 ,
      -2.83912292,  -2.13689355],
    [  0.57225191,  -3.61595657,   5.20454152,   0.25544555,
      -9.29180502,  -1.29562073,   5.16715666,   0.26788865,
      -8.40191228,  -5.30763609]]),
    array([[ 5.81272873e-01,  9.84053844e-01,  4.14704775e-01,
     -2.06282537e-01,  1.18202950e-01,  1.14484999e-01,
      8.40380544e-01,  5.08363894e-01,  5.37588582e-01,
      2.44707272e-01],
    [ 3.01574817e-01, -1.32886670e-01, -3.13403967e+00,
     -3.87593489e+00, -3.10878750e+00, -3.35675574e+00,
     -1.26763368e+00,  6.34694509e+00,  4.40952708e+00,
      1.16374496e+01],
    [ 6.53200820e-01,  2.66302770e+00, -4.31261312e-02,
     -1.06285184e+00, -3.38030677e-01, -1.59612424e+00,
      2.80010884e+00,  1.06808243e+00,  4.54052329e-01,
      1.10055991e+00],
    [ 9.19482614e-01,  1.69499555e+00,  2.34529072e-01,
     -1.40139244e+00,  6.41200971e-02, -1.42301822e+00,
      1.61575406e+00, -2.86045449e-02,  3.20115219e-01,
      9.77257452e-01],
    [ 5.69100739e-01,  3.66526968e+00, -2.23405404e-01,
     -1.68560951e+00, -2.89273399e-01, -1.70853181e+00,
      3.86678932e+00,  1.62492942e+00,  7.59519682e-01,
      2.13091006e+00],
    [ 3.59978064e-01, -3.77919630e+00, -3.05527610e+00,
      3.59905429e+00, -1.87094223e+00,  4.34955072e+00,
     -4.21266658e+00,  7.40291733e-01, -4.70227830e-01,
     -1.31723204e+00],
    [ 9.28081293e-01,  1.21953728e+00, -1.46091073e+00,
     -2.71667703e+00, -1.35261508e+00, -2.12949270e+00,
      5.71390571e-01,  3.34052454e+00,  2.15324028e+00,
      5.69361098e+00],
    [ 5.89909976e-01,  2.89380011e+00,  5.43733938e-03,
     -1.37688333e+00, -3.08207525e-01, -1.58274514e+00,
      3.05138276e+00,  1.34572353e+00,  4.32306328e-01,
      1.90604067e+00],
    [ 2.54356482e-01, -1.56212580e+01, -3.98764940e+00,
      8.40690689e+00, -1.18653703e+00,  1.04788097e+01,
     -1.39958146e+01, -5.40367957e+00, -7.21044500e+00,
     -9.69328112e+00]]),
     array([[  0.45369684,   2.20593905,  11.69994023,   4.19643201,
       2.80355976,   5.17745173,  -4.7942514 ,   5.51047876,
       4.52595547, -18.93013939],
    [  0.38346389,  -2.79365695, -11.09285132,  -4.13450582,
      -3.3947854 ,  -5.57727186,   4.67186938,  -6.27488213,
      -4.12466694,  18.94791879]])]
final cost is 0.000871182074484481
```

```
[[-0.6  -0.85]] predicted as [[9.99969058e-01 3.01420405e-05]]
[[ 0.75 -0.06]] predicted as [[3.76697413e-08 9.99999966e-01]]
```

The companion plots would be as shown in Fig. 1.



(a) Decision boundary in XOR dataset

(b) Decision boundary in blobs dataset

(c) Decision boundary in moons dataset

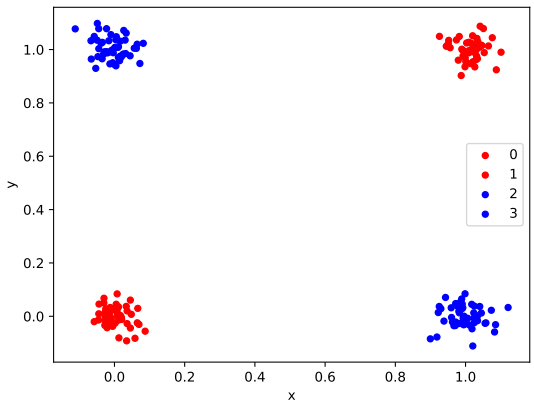
(d) Decision boundary in circles dataset

Figure 1: Expected results for the ANN feed forward + backprop implementation

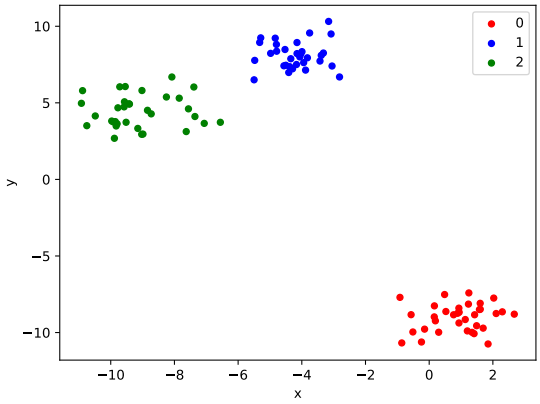
3 Datasets

You are provided with several datasets.

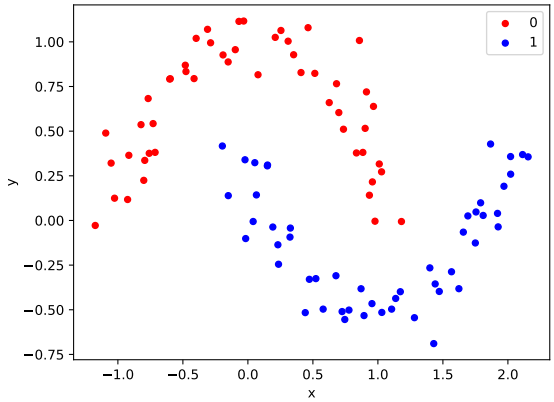
- XOR problem. Shown in Fig 2a.
- Blobs: Three separable clouds of data, shown in Fig. 2b.
- Moons: Two intertwined moon-shaped sets of data, shown in Fig. 2c.
- Circles: Two data classes nested in circles, shown in Fig. 2d.



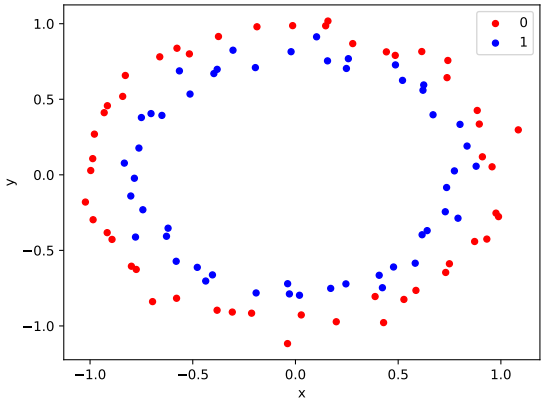
(a) XOR dataset



(b) Blobs dataset



(c) Moons dataset



(d) Circles dataset

Figure 2: Datasets

4 Further work

Prepare a short report (4 text pages top, could be more with cover, figures), where you present your findings during your implementation. After you implemented correctly your code, for this assignment, you need to play around with hyperparameter values that are set in the `launcher.py` file. Can you find smaller architectures that can be trained faster and with a similar error? **The report MUST include a section with the individual contributions from each team member.**

5 Notes

- Your code should be ready to be run by just executing the `launcher.py` script.
- Include everything (even datasets) in your submitted code so that it can be executed straightforwardly.
- **It will be penalized if your code can't be run directly.**

6 Deliverables

- Your implementation as a zip file with the source code.

  - Name your file as `A1_A2_A3-HW-02.zip` where A1, A2, A3 are the reg number (matrícula) of students.
  - Include the names of team members and student numbers as comments.
- Your report as a PDF.

A single submission per team is enough, there is no need for submitting more than once.