TC3020 Machine Learning
Assignment 3: Artificial Neural Networks

Dr. Rafael Pérez Torres

February 16th, 2021

Abstract

In this assignment, you will implement the FeedForward and Backpropagation steps of the training process of an Artificial Neural Network.

1 Description

Overall goal: Implement the training and prediction processes of an Artificial Neural Network. The implementation should offer the flexibility to create a variable number of hidden layers, any number of output neurons, and support for any number of n (features in the input). One of your implementations of LR will use regularization.

Considerations

- You can use the programming language of your choice. The only restriction is not to use any specialized library that could prevent you from implementing the core of the ANN FeedForward and Backpropagation algorithms. For example, if you are using Python, do not use `keras`, `tensorflow`, or `scikit`. Using `numpy` is fine though.
- Base your implementation on the material studied in class. It is also possible to use a different reference to develop your implementation, but in that case you MUST COMMENT THE CODE so as to clarify what is doing each function or code section.

2 Implementation requirements

2.1 Network architecture

The network architecture must be flexible, that is:

- The used dataset might have an arbitrary number of features (n).
- The number of hidden layers and neurons for these layers must be configurable. To simplify things you can set the same number of hidden neurons per hidden layer, but if you want to support different sizes that is fine too.
- The number of output neurons can be arbitrary, that is, your architecture should support multiclass classification.

2.2 Modularization

Assume you would be providing your code as a library for the benefit of the community, so that you are encouraged to modularise your code. This means that you should add parameters to your train function. Examples of parameters are: learning rate (α), regularization factor (λ), and number of epochs (*epochs*).

2.3 Steps

The train function is going to be using a forward and a backpropagation steps. This means you can modularize your code creating these two functions. Furthermore, to perform predictions, it will be handy to have the forward as a function.

Print the values of the cost function after each iteration/epoch, this will help you to ensure that your implementation is correct.

3 Dataset

You are provided with several datasets.

- XOR problem. This is similar to the example seen in class. As it is a simple, it will help you to perform a quick test of your implementation. Shown in Fig 1a.
- Manually defined datasets:
  - Blobs: Three separable clouds of data, shown in Fig. 1b.
  - Moons: Two intertwined moon-shaped sets of data, shown in Fig. 1c.
  - Circles: Two data classes nested in circles, shown in Fig. 1d.

Recall that for multiclass classification, you will need to create vector to encode the output. For instance in the blobs dataset, if an example $x^(i)$ is of class 1, its encoding should be $[0, 1, 0]$, and this means that in the output layer, the three neurons are expected to have that output as a group.

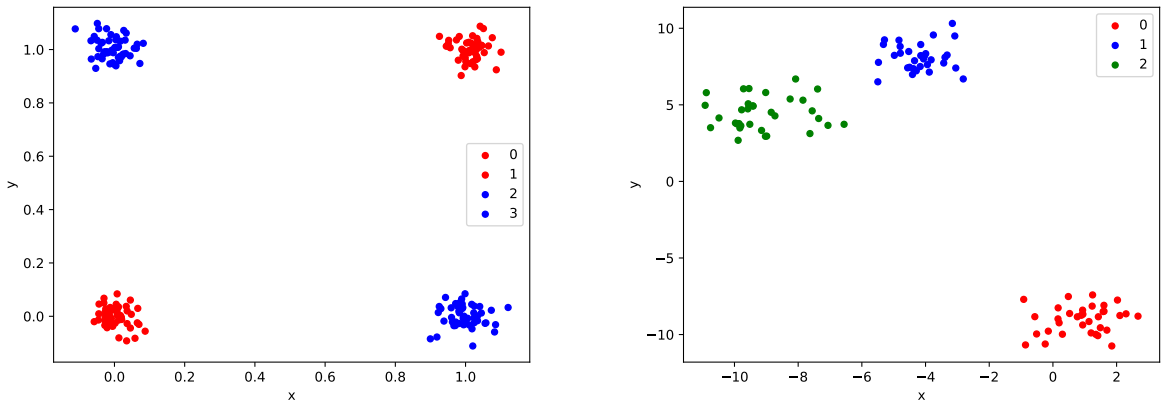
4 Further instructions

- As usual, play with the values of your parameters.
- It is ok to use a ML library **to verify / compare your implementation**, but remember the implementation is from scratch.
- It could be recommended to start with a fixed ANN architecture and then make the generalization to support other datasets. In this assignment the performance of the implementation is not evaluated. Nevertheless you might find that vectorized implementations are simpler to read and develop.
- If desired, you can follow an OOP implementation where the layers and weights are attributes of the network, etc.

4.1 Deliverables

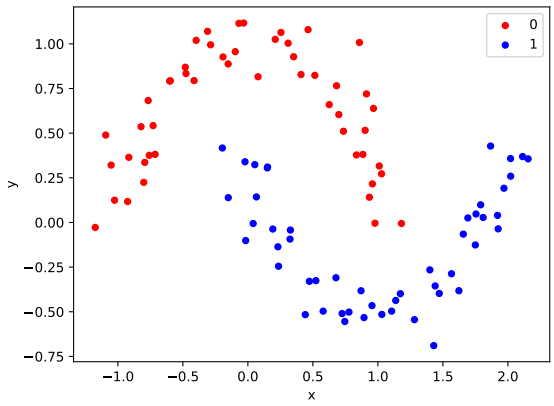
- A short report (2-4 pages is fine). This report should include your findings when playing around with parameter values, and predicting the output for input samples.
- You can also mention how many layers or hidden neurons were needed to achieve good values for correctly predicting classes. Screenshots (good quality please), or listing the console output to demonstrate the correctness of your implementation is welcome. For this assignment, the report can be in Spanish.
- Your implementation: you can upload a zip file with the source code or just share a link to a Github/Gitlab/etc repository.
  - If you share a repo, make sure you add a readme or describe your info in the repo.
  - If your language of choice is not JavaScript, Java, or Python, please give instructions in the report or readme regarding how to set up the environment to execute your implementation.

  Regardless of the programming language, **clearly** state what is the entry point for your implementation.

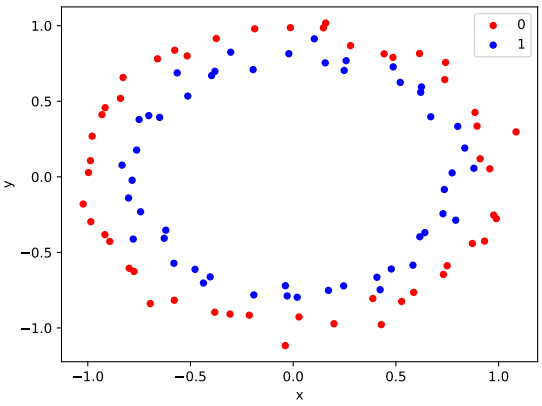


(a) XOR dataset

(b) Blobs dataset



(c) Moons dataset



(d) Circles dataset

Figure 1: Datasets

Recall that this assignment is in teams, the report must include the name and reg. number of the members. One upload is enough in Canvas, there is no need for all team members to upload material individually.