Laboratory: Artificial Neural Networks

Final Project (Spring 2019)

Apply the techniques you have learned to implement a neural network. Below are the steps you should follow.

It is noted that the work is done individually.

You have to upload in eclass, a .zip containing the .m files you've created, and a detailed report (PDF) with all the necessary figures, tables, and remarks.

Attention: In the report you should NOT embed the code you have made.

The purpose of the project is to build a neural network from scratch and train it.

Final submission date : 20/5/2019.

If you have any questions, please contact me via email: karampidis@staff.teicrete.gr

You can implement the neural network in Octave or Python. In both cases your code should be well commented.

Step 1

Load the haberman.rar file from eclass (in Appendix folder)

Information on this data set can be found here:
https://archive.ics.uci.edu/ml/datasets/Haberman%27s+Survival

Display the samples by using the gscatter[1] function (use the best two features).

Read carefully about the information is given to you and keep/remove/replace whatever you think is needed.

Step 2

Create training set and test set. Data separation will be done either with the builtin octave subset function or the basicsets function which is uploaded to eclass. Start with a separation rate of 70%-30% for training and test set respectively.

Pre-process your data so that they are in range [0 1]. The formula you will use is the following:

$$X_n = \frac{(X - Xmin)}{(Xmax - Xmin)} * (-1) + 1$$

Where:

X – value that should be normalized
Xn – normalized value
Xmin – minimum value of X
Xmax – maximum value of X

Step 3

Create a neural network of your choice (number of hidden layers, neurons in each layer etc.) which accepts as an input the features of each training example (from Step 1). Use the following parameters:

- Learning rate: 0.3
- Maximum number of epochs to train: 1000
- Performance goal: 1e-5
- Epochs between displays:10

Step 4

Present the trained neural network the test set and display the categorized patterns using the gscatter function. Compare and comment on your results in relation to the actual class.

Step 5

Optimize the neural network by adding or subtracting layers / neurons, by changing the number of epochs and record the results in a table for every change you have made.

Step 6

After completing all the steps, repeat the procedure (for the architecture that gave the best results) with split rates (for dataset): 50%-50%, 60%-40%, 80%-20%, 90%-10% and list the results in a table. What do you conclude?

Step 7

Combining all the outcomes of the above steps and having in mind whatever you have learned this semester in this course, what do you think that is the most essential part of constructing a neural network? Give a general conclusion

1. To use the gscatter function, you must load the nan package in the Octave. More about this function can be found here: https://octave.sourceforge.io/nan/function/gscatter.html