Laboratory: Artificial Neural Networks

Final Project (Spring 2020)

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

It is noted that the work should be done individually.

You have to upload in eclass, a .zip containing the .ipynb 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: 31/5/2020.

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

Step 1

In Appendix folder (in eclass) you will find four files concerning different datasets. Choose one of them and load it into your jupyter notebook.

Information on these data sets can be found here:

https://archive.ics.uci.edu/ml/datasets/Haberman%27s+Survival
https://archive.ics.uci.edu/ml/datasets/banknote+authentication
http://archive.ics.uci.edu/ml/datasets/Teaching+Assistant+Evaluation
http://archive.ics.uci.edu/ml/datasets/Glass+Identification

Visualize the samples using a scatter plot (use the pandas and/or seaborn packages methods).

Read carefully about the information is given to you and remove and / or replace whatever information you think is not needed.

Step 2 (optional)

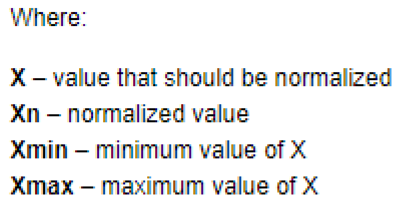
Check for outliers and remove them if they exist.

Check if there are missing values and replace them with the mean value (of the respective column and category) or remove them (justify your decision).

Step 3

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

You can also try MinMaxScaler from scikit-learn to see if you have better results.

Create training set and test set. Data separation will be done with the built-in scikit-learn subset function. Start with a separation rate of 70%-30% for training and test set respectively.

Step 4

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: 3000
- Performance goal: 1e-5
- Epochs between displays:100

Step 5

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

Step 6

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 7

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 8

Create a Kohonen network and implement clustering to the data (unsupervised learning).

Apply everything you have learned from the laboratory about Kohonen networks.

Step 9

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.

Good Luck !!!