Nancy Espinosa Castillo - A01206782
Joel Alexis Bustamante Calvario - A00226215

Perceptron / Neural Networks

Artificial Neural Networks are processing devices that are loosely modeled after the neuronal structure of the cerebral cortex but on much smaller scales. According with Maureen Caudill, these are computing systems made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs.

In order to test an ANN in Weka the team used a data set that represents characteristics of different subspecies of flowers. For search a property data set, we specified the next three facts:

- A data set Multivariate, to have a data set with more of one variable.
- A data set with Classification, to helps to ensure the discrete attributes.

With this features we selected the data set of Iris [1] that contains 150 instances and 5 attributes.

The attributes of the set are the next:

- Sepal length
- Sepal width
- Petal length
- Petal width
- Class with the values Iris Setosa/Versicolour/Virginica

Multi-layer perceptron

To build the ANN, the team decided out of the 150 vectors, 70% for training and 30% for testing. The following are the results of the ANN of Iris data set.

```
Relation:     iris
Instances:    150
Attributes:   5
              sepal length
              sepal width
              petal length
              petal width
              class
Test mode:    split 70.0% train, remainder test

=== Classifier model (full training set) ===
Sigmoid Node 0
    Inputs    Weights
    Threshold    -3.5015971588434014
    Node 3    -1.0058110853859945
    Node 4    9.07503844669134
    Node 5    -4.107780453339234
Sigmoid Node 1
    Inputs    Weights
    Threshold    1.0692845992273177
    Node 3    3.8988736877894024
    Node 4    -9.768910360340264
    Node 5    -8.599134493151348
Sigmoid Node 2
    Inputs    Weights
    Threshold    -1.007176238343649
    Node 3    -4.2184061338270356
    Node 4    -3.626059686321118
    Node 5    8.805122981737854
Sigmoid Node 3
    Inputs    Weights
    Threshold    3.382485556685675
    Attrib sepal length    0.9099827458022276
    Attrib  sepal width    1.5675138827531276
    Attrib  petal length    -5.037338107319895
    Attrib  petal width    -4.915469682506087
Sigmoid Node 4
    Inputs    Weights
    Threshold    -3.330573592291832
    Attrib sepal length    -1.1116750023770083
    Attrib  sepal width    3.125009686667653
    Attrib  petal length    -4.133137022912305
    Attrib  petal width    -4.079589727871456
Sigmoid Node 5
    Inputs    Weights
    Threshold    -7.496091023618089
    Attrib sepal length    -1.2158878822058787
    Attrib  sepal width    -3.5332821317534897
    Attrib  petal length    8.401834252274096
    Attrib  petal width    9.460215580472827
Class Iris-setosa
    Input
    Node 0
Class Iris-versicolor
    Input
    Node 1
Class Iris-virginica
    Input
    Node 2

Time taken to build model: 0.11 seconds
```

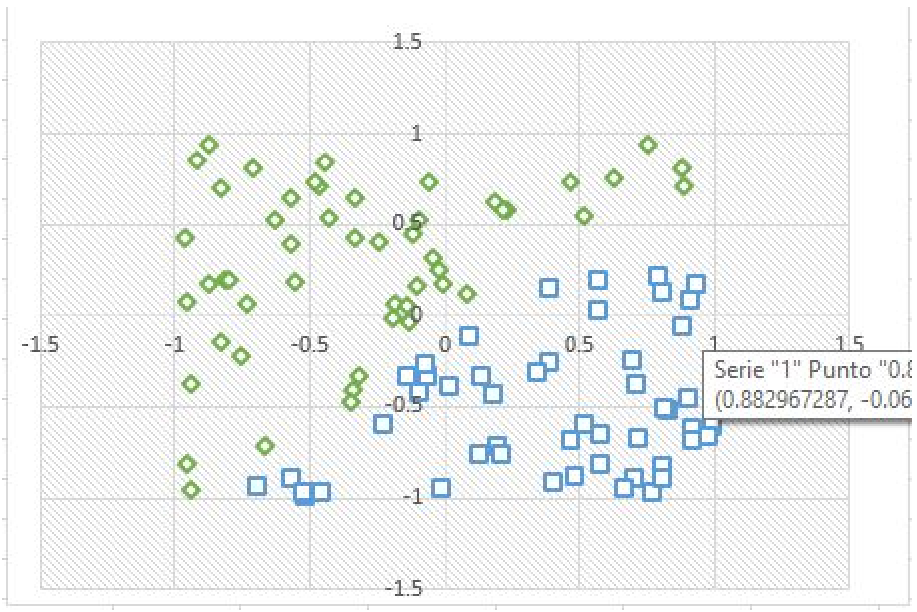
```
=== Evaluation on test split ===

Time taken to test model on test split: 0 seconds

=== Summary ===

Correctly Classified Instances          44               97.7778 %
Incorrectly Classified Instances         1                2.2222 %
Kappa statistic                      0.9666
Mean absolute error                      0.024
Root mean squared error                    0.1153
Relative absolute error                  5.3891 %
Root relative squared error              24.4455 %
Total Number of Instances                45

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     Iris-setosa
                 1,000    0,034    0,941      1,000    0,970      0,953    0,998     0,996     Iris-versicolor
                 0,933    0,000    1,000      0,933    0,966      0,950    0,998     0,996     Iris-virginica
Weighted Avg.    0,978    0,012    0,979      0,978    0,978      0,967    0,998     0,997

=== Confusion Matrix ===

  a  b  c   <-- classified as
 14  0  0 |  a = Iris-setosa
  0 16  0 |  b = Iris-versicolor
  0  1 14 |  c = Iris-virginica
```

Conclusions

After testing with this data set with 70% split rate, we can see that the error is 2.2% with one instance incorrectly classified. Also, when we tested with 80% split rate, the error increase to 3.3% and with 90%, this error increase to double. So we conclude that the error increase slower from 70% to 80% where between this percentages,it converges

Part 1

Scatter plot of the training set of the linearly_separable



Part 2

Perceptron and Artificial Neural Networks

A perceptron is the simplest mathematical model of a neuron and it can learn to predict the AND, OR and NOT functions in a quick way, because are linearly separable due the 4 possible outputs of each function and therefore can be learned. However, a single perceptron cannot learn XOR function since it requires at least two lines to separate classes (0 and 1) and it is necessary one additional layer of perceptrons to learn but this is more slower.

On the other hand, ANN gives the possibility of use an arbitrary approximation function mechanism that learns from observed data but its use is more complicated. So that ANNs are good for online learning and applications of large datasets.Also, most of the applications of ANN consist in make a recognize of patterns, such as: look for a pattern in a series of examples, sort patterns, starting from one distorted.

ANNs are worth implementing given a complex problem, for example, if the time the network takes to learn how to predict is lower than the time it would take a human to do a classification of data by hand, then it is worth it.

Despite their flexibility, ANNs are not the best option when a problem is far too complex because this can lead to overfitting, and an overfitted network would be useless.

References

1. https://archive.ics.uci.edu/ml/datasets/Iris