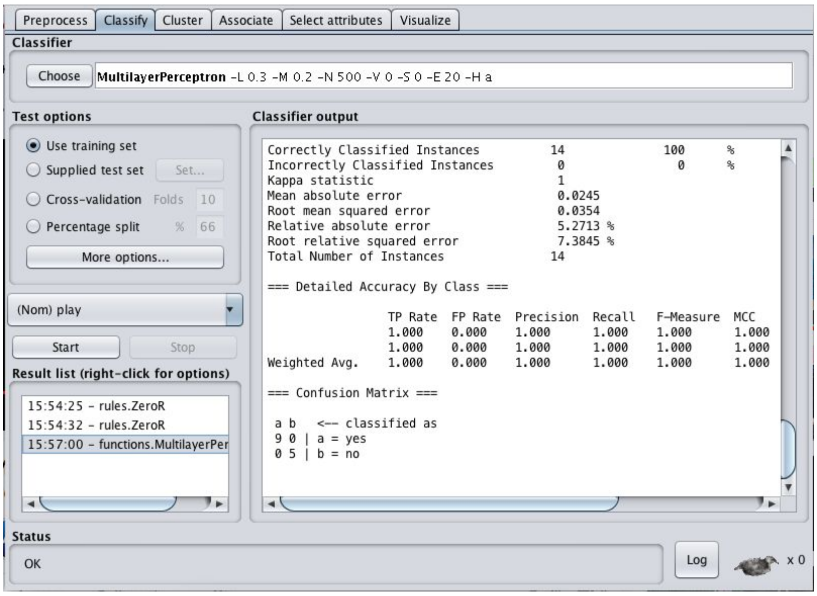
Raúl García Furió A01205785
Ana M. González Montoya A01204110

Report 7

Instructions:
Write a report using the comparisons collected by training the ANN with different parameters. Include an image of the network and the parameters used as well as the error and time it took to train each network. Write a brief reflection about what you think is happening in the different ANNs. Other interesting aspects you could include in your reflection are: Explanations as to what are ANNs good for. Where would you use them? Are they worth the effort implementing or not? What kinds of problems do they not solve?

When we train in weka de ANN networks we start without the knowledge of what we were doing.We change the parameters in order to understand how it works and we discovered that when you chance the hidden layers the ANN starts to be more complicated and with more and more layers also the percentage of error increase and change every time we change the form we train the network. [images at the end]

We discovered that ANNs are good to classify information, for example, in our challenge we are using ANNs to classify numbers, the user get the input of a number and the ANNs decide which number is.A big problem with the ANN is that, when they are complex, is difficult to train to get a good result and less percentage of error.



Classifier

MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

Test options

● Use training set
○ Supplied test set
○ Cross-validation Folds 10
○ Percentage split % 66

(Nom) play

Result list (right-click for options)

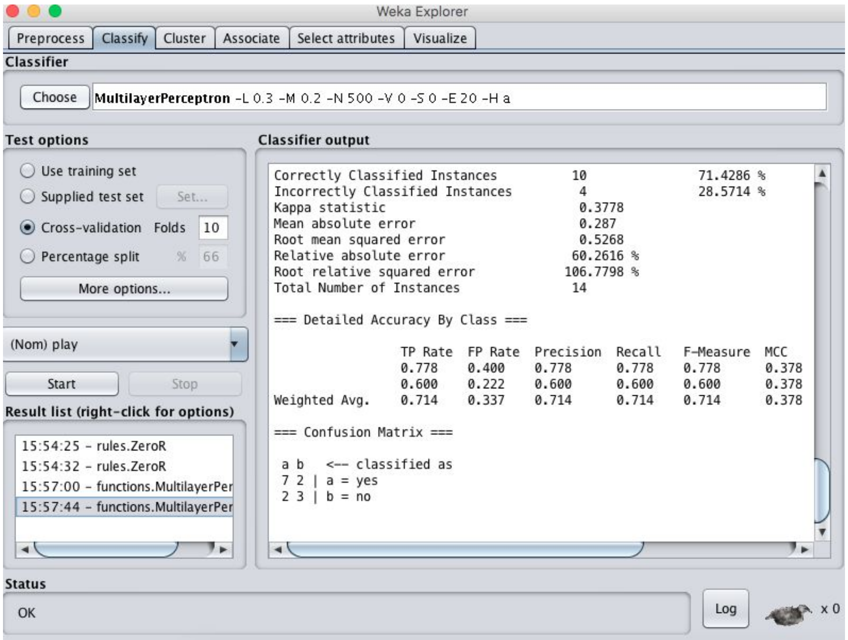
15:54:25 - rules.ZeroR
15:54:32 - rules.ZeroR
15:57:00 - functions.MultilayerPer

Classifier output

```
Correctly Classified Instances          14              100      %
Incorrectly Classified Instances         0                0      %
Kappa statistic                          1
Mean absolute error                      0.0245
Root mean squared error                  0.0354
Relative absolute error                  5.2713 %
Root relative squared error              7.3845 %
Total Number of Instances               14

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC
                 1.000    0.000    1.000      1.000    1.000      1.000
                 1.000    0.000    1.000      1.000    1.000      1.000
Weighted Avg.    1.000    0.000    1.000      1.000    1.000      1.000

=== Confusion Matrix ===

 a b   <-- classified as
 9 0 | a = yes
 0 5 | b = no
```

Status

OK



Classifier

MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

Test options

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) play

Result list (right-click for options)

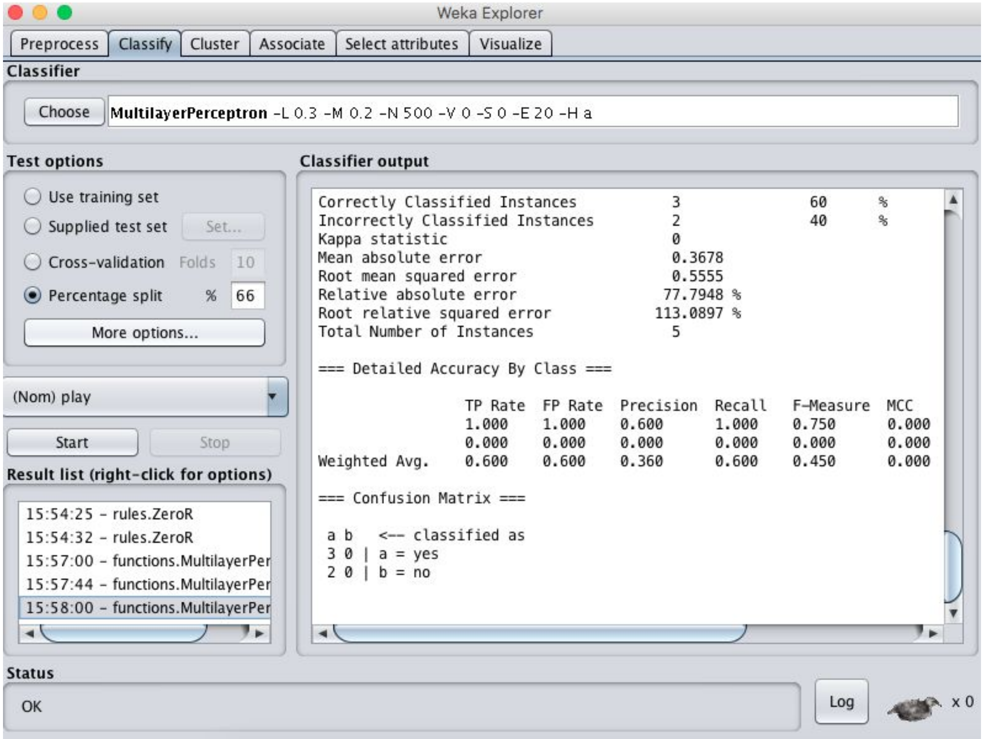
15:54:25 - rules.ZeroR
15:54:32 - rules.ZeroR
15:57:00 - functions.MultilayerPer
15:57:44 - functions.MultilayerPer

Classifier output

```
Correctly Classified Instances          10               71.4286 %
Incorrectly Classified Instances         4               28.5714 %
Kappa statistic                          0.3778
Mean absolute error                      0.287
Root mean squared error                  0.5268
Relative absolute error                 60.2616 %
Root relative squared error            106.7798 %
Total Number of Instances               14

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC
                 0.778    0.400    0.778      0.778    0.778      0.378
                 0.600    0.222    0.600      0.600    0.600      0.378
Weighted Avg.    0.714    0.337    0.714      0.714    0.714      0.378

=== Confusion Matrix ===

 a b   <-- classified as
 7 2 | a = yes
 2 3 | b = no
```

Status

OK



Classifier

MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

Test options

○ Use training set
○ Supplied test set
○ Cross-validation Folds 10
● Percentage split % 66

(Nom) play

Result list (right-click for options)

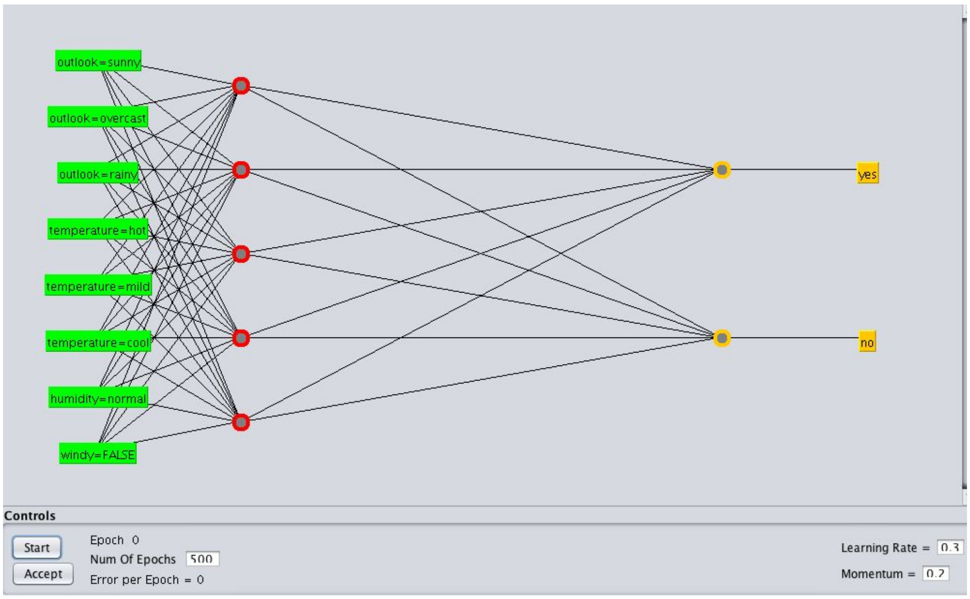
15:54:25 - rules.ZeroR
15:54:32 - rules.ZeroR
15:57:00 - functions.MultilayerPer
15:57:44 - functions.MultilayerPer
15:58:00 - functions.MultilayerPer

Classifier output

```
Correctly Classified Instances           3               60      %
Incorrectly Classified Instances         2               40      %
Kappa statistic                          0
Mean absolute error                      0.3678
Root mean squared error                  0.5555
Relative absolute error                 77.7948 %
Root relative squared error            113.0897 %
Total Number of Instances                5

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC
                 1.000    1.000    0.600      1.000    0.750      0.000
                 0.000    0.000    0.000      0.000    0.000      0.000
Weighted Avg.    0.600    0.600    0.360      0.600    0.450      0.000

=== Confusion Matrix ===

 a b   <-- classified as
 3 0 | a = yes
 2 0 | b = no
```

Status

OK



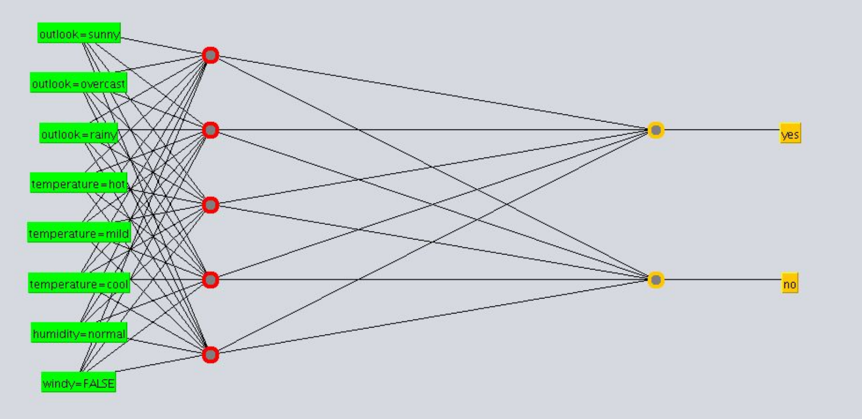
Epoch 0

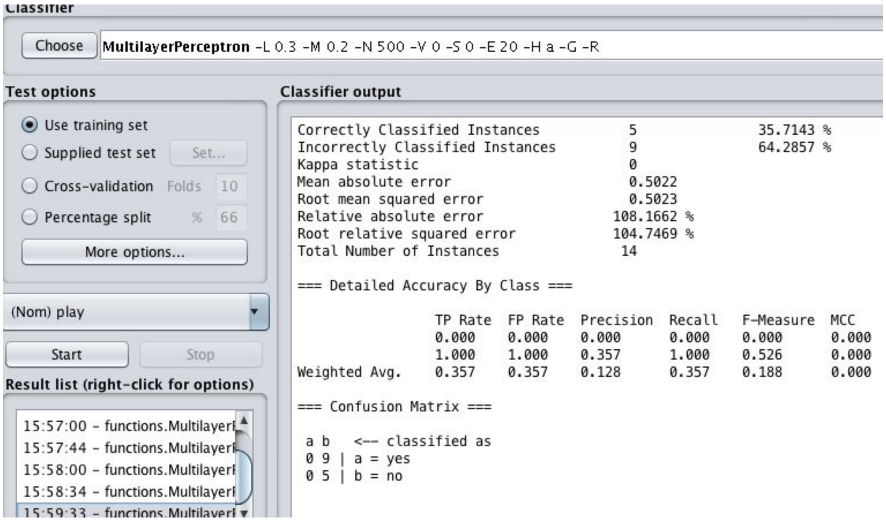
Num Of Epochs 500

Error per Epoch = 0

Learning Rate = 0.3

Momentum = 0.2





MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a -G -R

● Use training set
○ Supplied test set Set...
○ Cross-validation Folds 10
○ Percentage split % 66

(Nom) play

Result list (right-click for options)

15:57:00 - functions.MultilayerP
15:57:44 - functions.MultilayerP
15:58:00 - functions.MultilayerP
15:58:34 - functions.MultilayerP
15:59:33 - functions.MultilayerP

```
Correctly Classified Instances           5               35.7143 %
Incorrectly Classified Instances         9               64.2857 %
Kappa statistic                          0
Mean absolute error                      0.5022
Root mean squared error                  0.5023
Relative absolute error                108.1662 %
Root relative squared error            104.7469 %
Total Number of Instances               14

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC
                 0.000    0.000    0.000      0.000    0.000      0.000
                 1.000    1.000    0.357      1.000    0.526      0.000
Weighted Avg.    0.357    0.357    0.128      0.357    0.188      0.000

=== Confusion Matrix ===

 a b   <-- classified as
 0 9 | a = yes
 0 5 | b = no
```