Neural Network

Package: AMORE (“A MORE flexible neural network package”)

The aim of the task: create neural network which classifies flats to group 1 or 2 depending on their characteristics

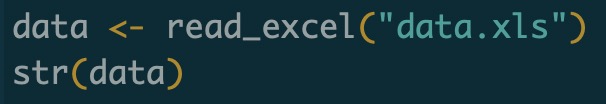
First I’ve artificially generated the data using **Excel** and function **RANDBETWEEN(1;8)** to create 100 rows with numbers from 1 to 8 in 4 columns: Large(L), Cheap(C), Armed(A) and Near the center city(NC).

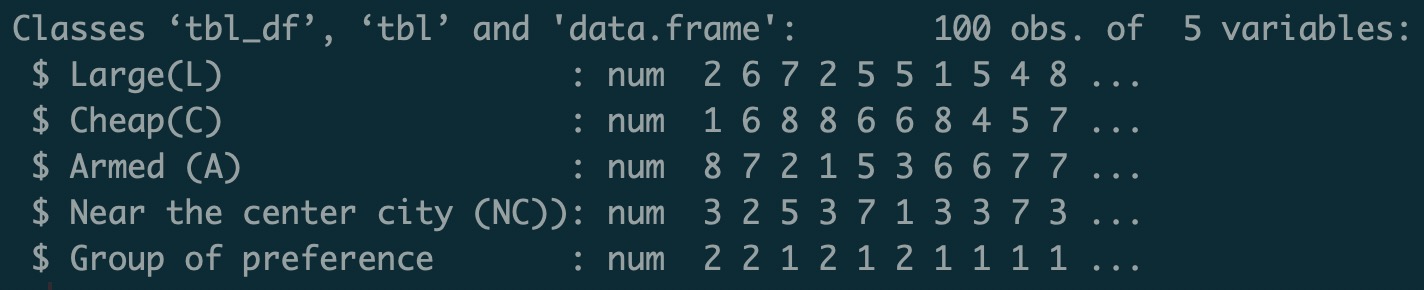
Then I’ve created a column “Group of reference” which takes values 1 or 2 depending upon the values in 4 columns. To do it I’ve used function **IF**:

A screenshot of a cell phone

Description automatically generated

Next step: loading the data and checking its structure to make sure that all columns were loaded as numeric:





Then I’ve checked the number of occurrences of each category:

A close up of a logo

Description automatically generated

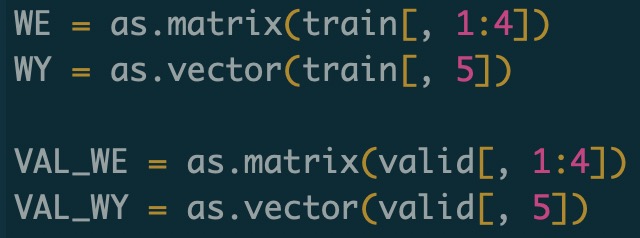
So, now I know that 51 flats out of 100 were classified to group 2 and 49 to group 1

Dividing data on training and validation sets:

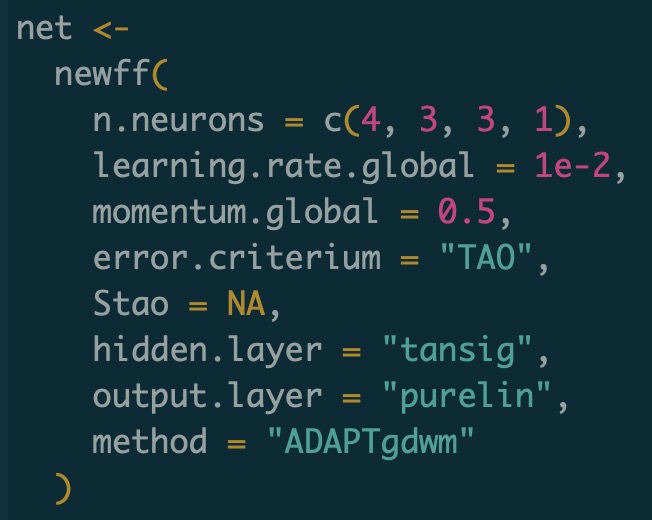
A close up of a sign

Description automatically generated

Then constructing matrices (input data as matrix with 4 columns) and vectors (values of “Group of reference”) for the model:



Creating variable which contains settings of the network:



Description of train function parameters from documentation:

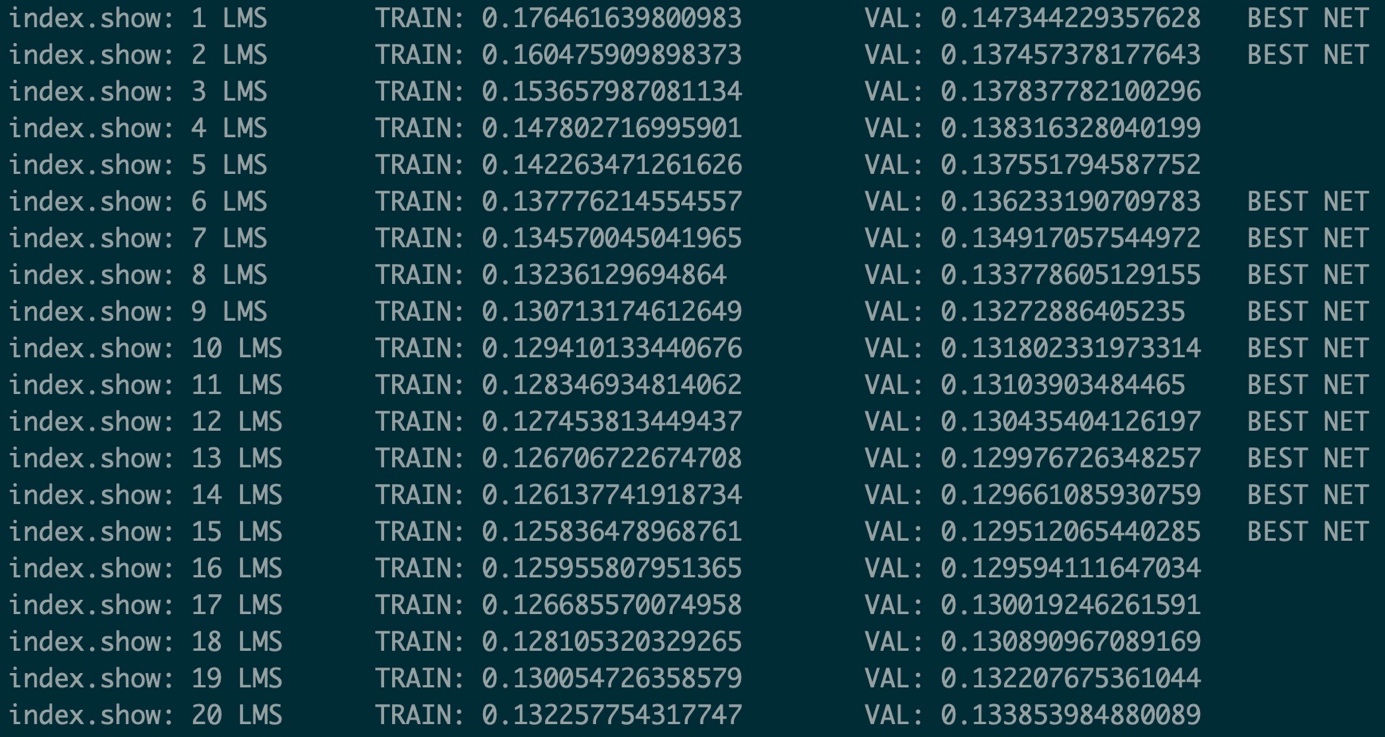
|  |  |
| --- | --- |
| n.shows | Number of times to report (if report is TRUE). The total number of training epochs is n.shows times show.step. |
| show.step | Number of epochs to train non-stop until the training function is allow to report. |

Now training the network (2000 epochs):

A close up of a sign

Description automatically generated

Every 100th iteration (epoch) of the model:

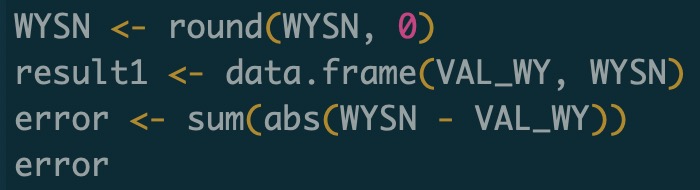


It is seen here - at some point, after 200th epoch model starts overfitting.

Using **sim** function generate the output from network:



Now comparing results from network with the original ones:

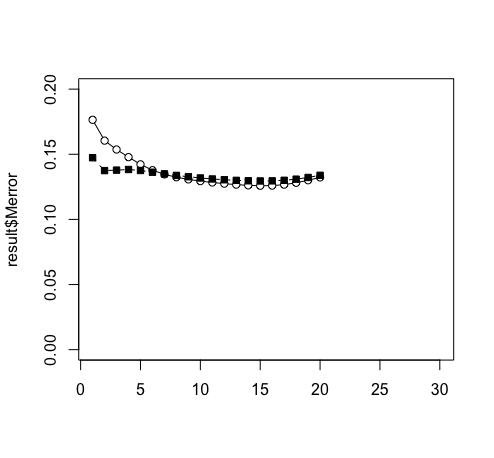


A close up of a logo

Description automatically generated

So, 5 cases out of 30 were classified incorrectly.

Plot shows the same - at some point model starts overfitting:



Now I’m going to adjust some parameters of the model to increase its accuracy and avoid overfitting:

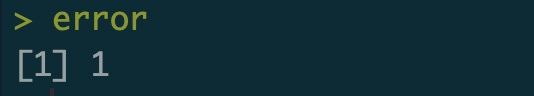
Increasing number neurons inside network:



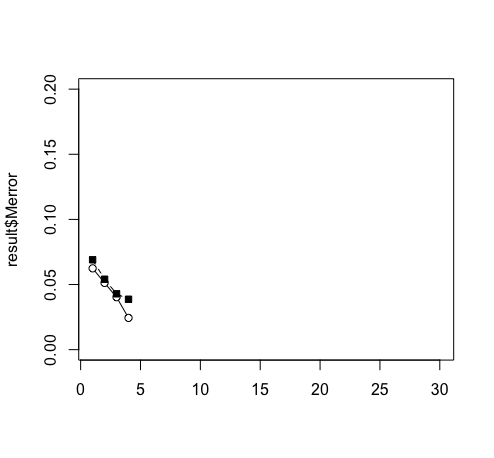
Decreasing number of epochs:



Network accuracy increased to 1 error instead of 5:



Now we can see from the plot that there is no overfitting and error is decreasing over the time



**Conclusion:** considering the fact that our dataset is quite small (only 100 observations) I’ve decided to decrease the number of epochs from 2000 to 400 and increase the number of neurons on 2 hidden layers from 3 to 10. As a result – error decreased from 5 to 1 and model is not overfitting anymore.