Juan Carlos Gama Flores A01201796

Intelligent Systems by Ruben Stranders

# Perceptron/Neural Networks Laboratory

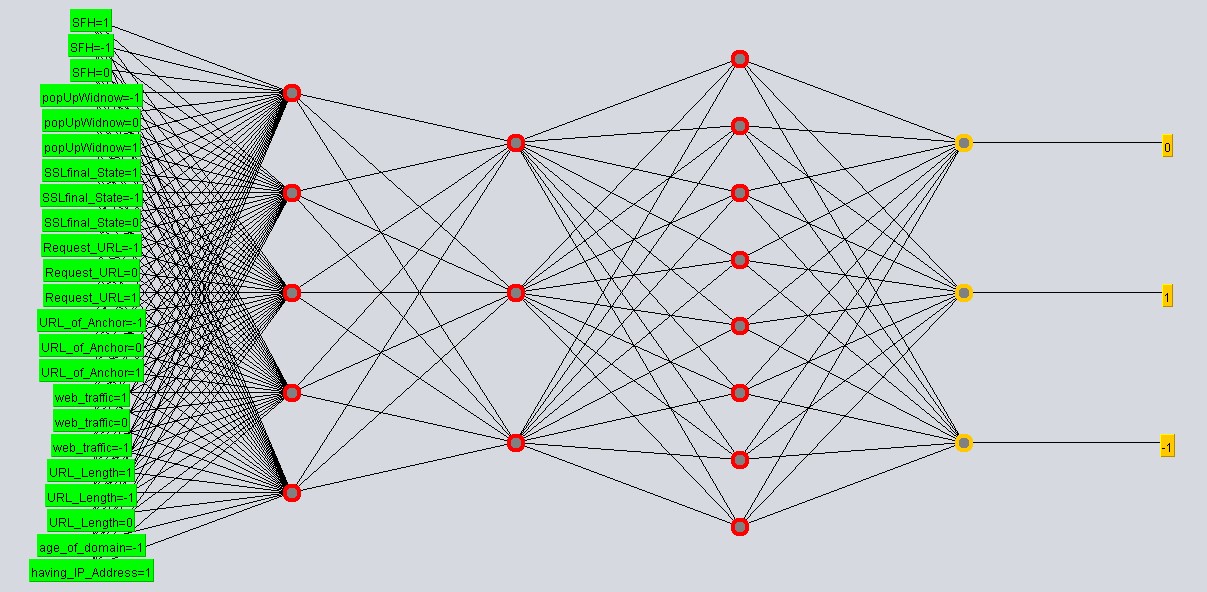
## Linearly separable example scatter plot

This plot proves that the data given is effectively linearly separable, thus a perceptron (neural network of one hidden layer and one output) can separate data and give solutions for given inputs.

## Artificial Neural Networks Analysis with Weka

The set of data used for this analysis is called Phishing data. It is compound of 10 attributes (inputs) and 1353 instances.

### Neural Network One



Learning rate = 0.3

Momentum = 0.2

Epochs = 500

Data used for training

In 500 epochs (iterations) the error per epoch decreased to 0.039

Correctly classified instances = 93.34%

Incorrectly classified instances = 6.65%

Time taken to test model on training data: 0.03 seconds

50% of data for training and 50% for testing

For testing:

In 500 epochs (iterations) the error per epoch decreased to 0.041

Correctly classified instances = 85.79%

Incorrectly classified instances = 14.20%

Time taken to test model on test split: 0.01 seconds

Conclusion: The amount of training data is little to expect a result near to 99.99% of correct instances classified when testing.

Changing learning rate = 0.8

Data used for training

In 500 epochs (iterations) the error per epoch decreased to 0.041

Correctly classified instances = 92.23%

Incorrectly classified instances = 7.76%

Time taken to test model on training data = 0.01 seconds

50% of data for training and 50% for testing

For testing:

In 500 epochs (iterations) the error per epoch decreased to 0.044

Correctly classified instances = 86.53%

Incorrectly classified instances = 13.46%

Time taken to test model on test split: 0.01 seconds

Conclusion: Have increased learning rate helped to reduce error (incorrectly classified instances) when testing. Time taken for training decreased from last prove, this must mean that for a higher learning rate the training algorithm (forward and backward propagation) is less “strict”.

Changing epochs = 1000 (from original NN parameters)

Data used for training

In 500 epochs (iterations) the error per epoch decreased to 0.038

Correctly classified instances = 93.05%

Incorrectly classified instances = 6.94%

Time taken to test model on training data: 0.02 seconds

50% of data for training and 50% for testing

For testing:

In 500 epochs (iterations) the error per epoch decreased to 0.040

Correctly classified instances = 85.94%

Incorrectly classified instances = 14.05%

Time taken to test model on test split: 0.02 seconds

Conclusion: Have increased epochs decreased very little error. Time taken for testing doubled, as the epochs (this is a very expected time result).

20% of data for training and 80% for testing (from original NN)

For testing:

In 500 epochs (iterations) the error per epoch decreased to 0.026

Correctly classified instances = 82.99%

Incorrectly classified instances = 17.00%

Time taken to test model on test split: 0.02 seconds

Conclusion: As expected, error increased compared to when it is 50% training and 50% testing, because NN had less data to learn. That was a behavior we have seen since we reduced 100% training to 50% training and 50% testing, that is why this was expected. Considering the % of data for testing from 50 to 80, the time taken doubled from 0.01 to 0.02. An expected result, just because more data was used for testing.

80% of data for training and 20% for testing (from original NN)

For testing:

In 500 epochs (iterations) the error per epoch decreased to 0.041

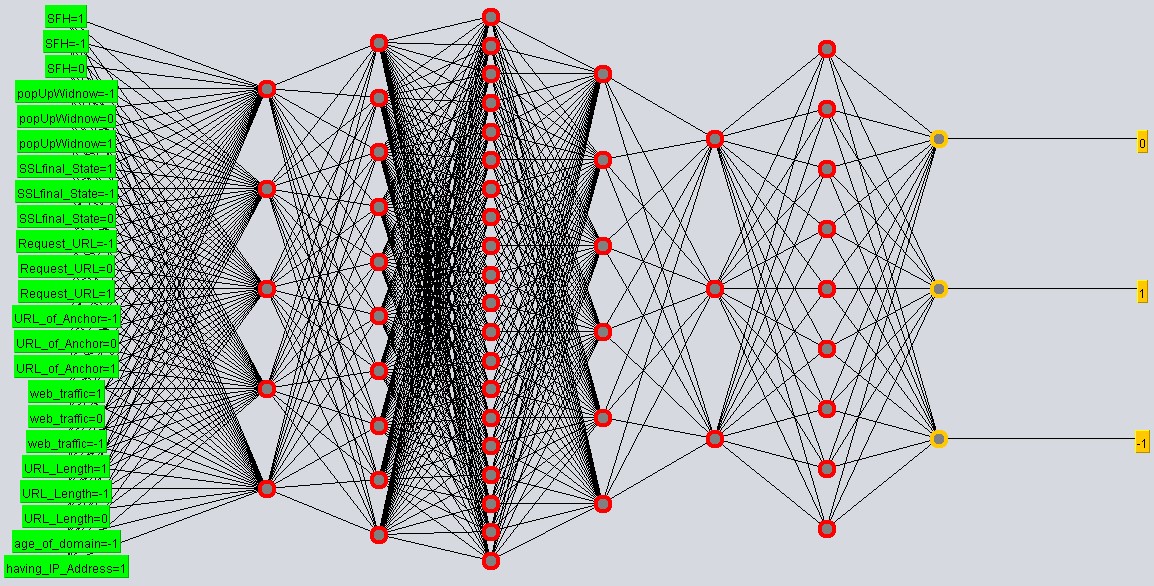
Correctly classified instances = 87.82%

Incorrectly classified instances = 12.17%

Time take to test model on test split: 0.01seconds

Conclusion: Error decreased a little compared to when It is 50%/50%. This is a very good NN because neither in the 20%/80% nor in the 80%/20%, it went overfitted.

### Neural Network Two



Learning rate = 0.3

Momentum = 0.2

Epochs = 500

Time taken to test model on training data (100% of data for training): 0.07 seconds

90% of data for training and 10% for testing

For testing:

Correctly classified instances = 54.81%

Incorrectly classified instances = 45.18%

Time taken to test model on test split: 0.01 seconds

Conclusion: Obviously this NN is being overfitted. Too much data for training, so when it is tested, results are wrong. In this case almost 50% correct and 50% wrong. Time taken is very little because it was used just 10% of the data. Even for this complex NN, using 10% of data for testing is fast =

0.01 seconds. On the other hand, when using 100% of data for training, time increase from 0.03 to 0.07 seconds between the less complex and the more complex NN, same parameters, different NN.