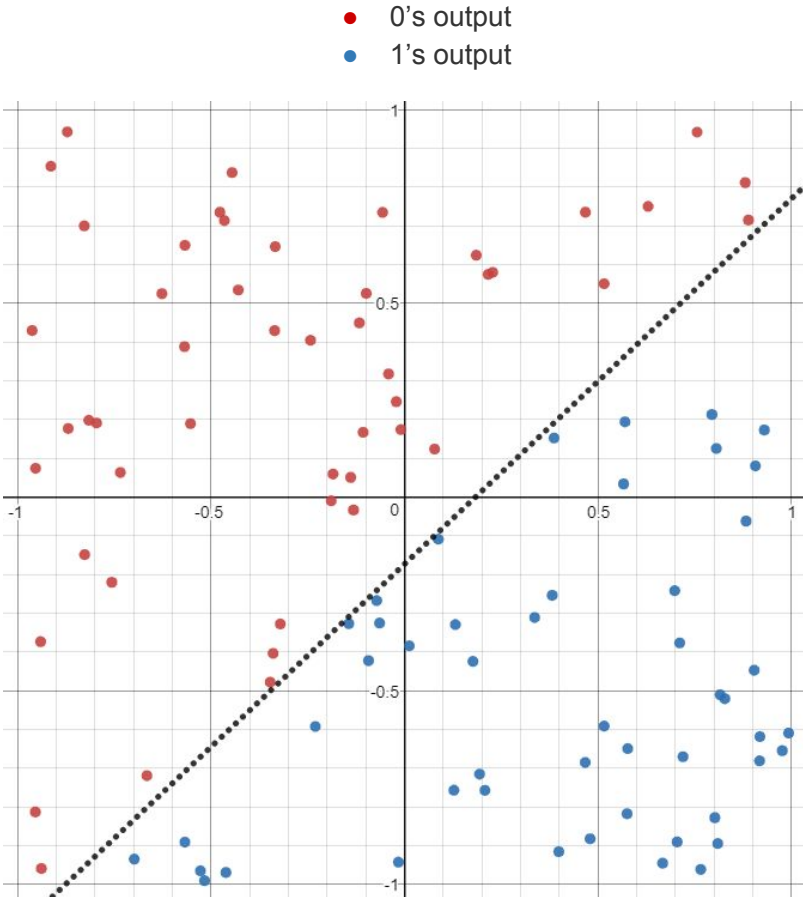
Carlos Roberto Cueto Zumaya A01209474
Moises Alexander Gama Espinosa A00817566

ANN

Part 1

Create a scatter plot of the training set of the `linearly_separable` example in the tests section. Use different colors or symbols for the two classes. Draw the line that the perceptron algorithm found. Include the mathematical equation of the line in your report.

$$\frac{1.23813445042359 \cdot x - 0.22726467547901972}{1.3122534872693066}$$

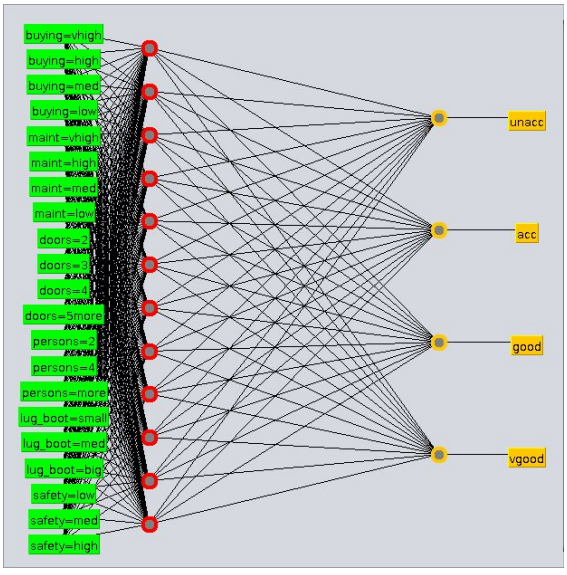
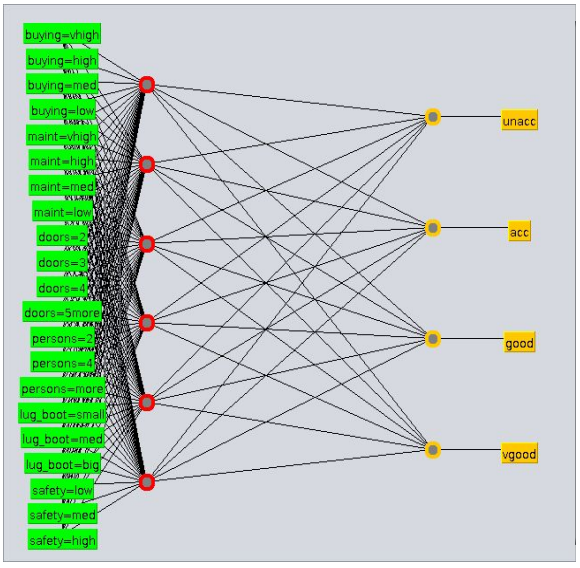


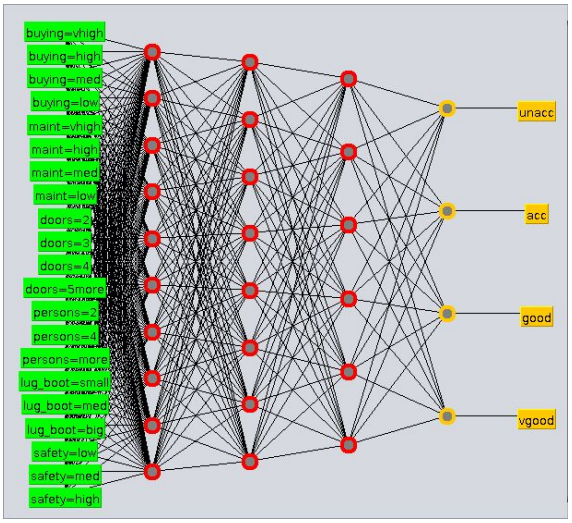
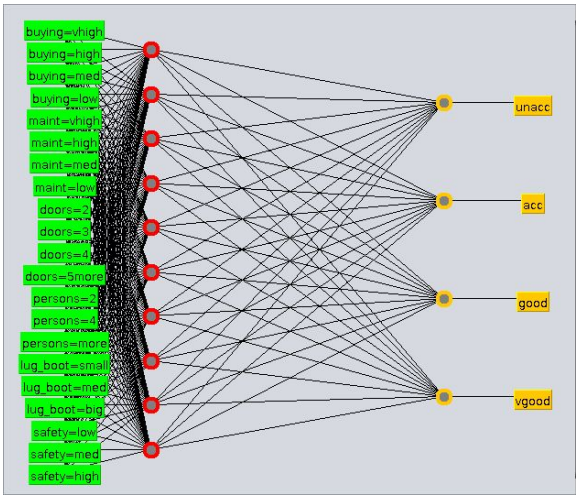
Part 2

Write an analysis of training the ANN with different parameters. Include an image of the network, the parameters used, and the error and time it took to train each network. Write a brief reflection about what you think is happening in the different ANNs, and why the ANN is behaving like it is. How did the choice of parameters influence the behavior of the ANN?

For the dataset, we used the one that gave us problems on the Decision Trees lab about classification of cars based on the security, price and performance and here the data was managed correctly mainly because an ANN is capable of process more input fields.

We played with the number of hidden layers, the learning rate and epochs and there were a lot of changes primarily the error per epoch, correct/incorrect guesses and the time it took to train and build, the key points we discover were that, if we increase the learning rate the relation of correct/wrong classifications vary a lot but if we decrease it there is not a big difference only if we keep the number of layers but if we increase the amount of layers or number of epochs the training and build time slows down the process a lot and sometimes the error decreases so the correctly classified instances increase on the last test we decided to decrease all the parameters and it was the worst performer as there were more incorrect classifications.

1	2
	

3 , 4, 5	6
	

Test	Learning rate	Epochs	Hidden layers	Build time	Training time	Correct classifications	Incorrect classifications
1 (nothing changed)	0.3	500	a	4.36	0.02	1728	0
2 (hidden layers)	0.3	500	6	36.62	0.01	488	1240
3 (Increase epochs)	0.3	800	10,8,6	151.43	0.01	1728	0
4 (increase learning rate)	0.8	800	10,8,6	36.71	0.01	1724	4
5 (decrease learning rate and hidden layers)	0.01	800	10, 8	48.5	0.03	1728	0

6 (decrease learning rate, epochs and only 1 hidden layer)	0.01	300	10	8.2	0.01	65	1663

Other interesting aspects you could include in your reflection are:

- Explanations as to what are ANNs good for.

  They are useful where the data is linearly separable because the algorithm works best with that type of data, for example, the fields where they are used are predictions or recognitions, the ANN identify and classify in different classes the object appearing in a image or predict functions approximations or data analysis.

- Where would you use them?

  We believe that robotic vision is a good example of where we can use ANN, there are a lot of robot manufacturers used to keep costs down and improve the product quality and they need to recognize parts on a line assembly for an engine on a car production or the facility environment to navigate by itself to deliver the parts used on a certain station.

- Are they worth the effort implementing or not?

  Neural networks are a big step the computational power is big and it analyse data and gather key information that for example helps business to grow, maybe in the past there were only for experts but right now there are a lot of libraries like tensorflow or keras that make a little bit easier the implementation but they are not for everyone at least you need to know what you are doing, have an idea of what a neural network is and obviously know how to code, but the thing is that for real applications they are a tool that improves the solution and it's totally worth it to learn them and implement them if you want a really something powerful.

- What kinds of problems do they not solve?

  Its difficult for them to work with unlabeled data or data that is not linearly separable.