



**INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE
MONTERREY**

Artificial Intelligence

Report Lab 6

Team:

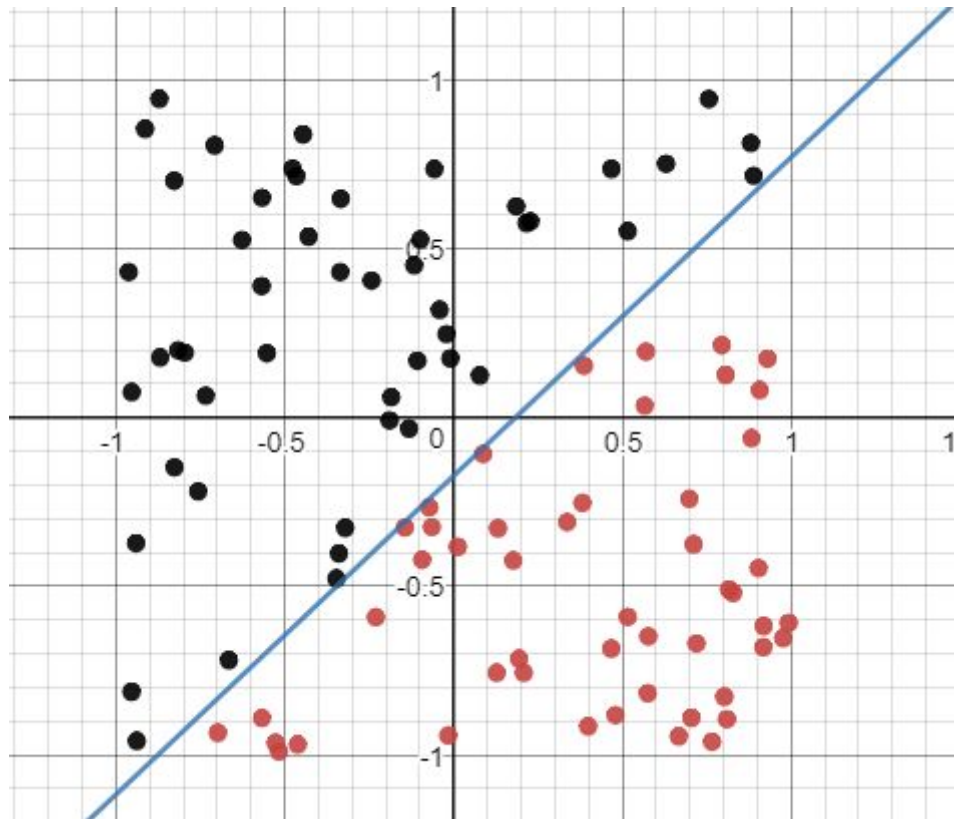
| | |
|----------------------------|-----------|
| Alan Kuri García | A01204805 |
| Shara Teresa González Mena | A01205254 |

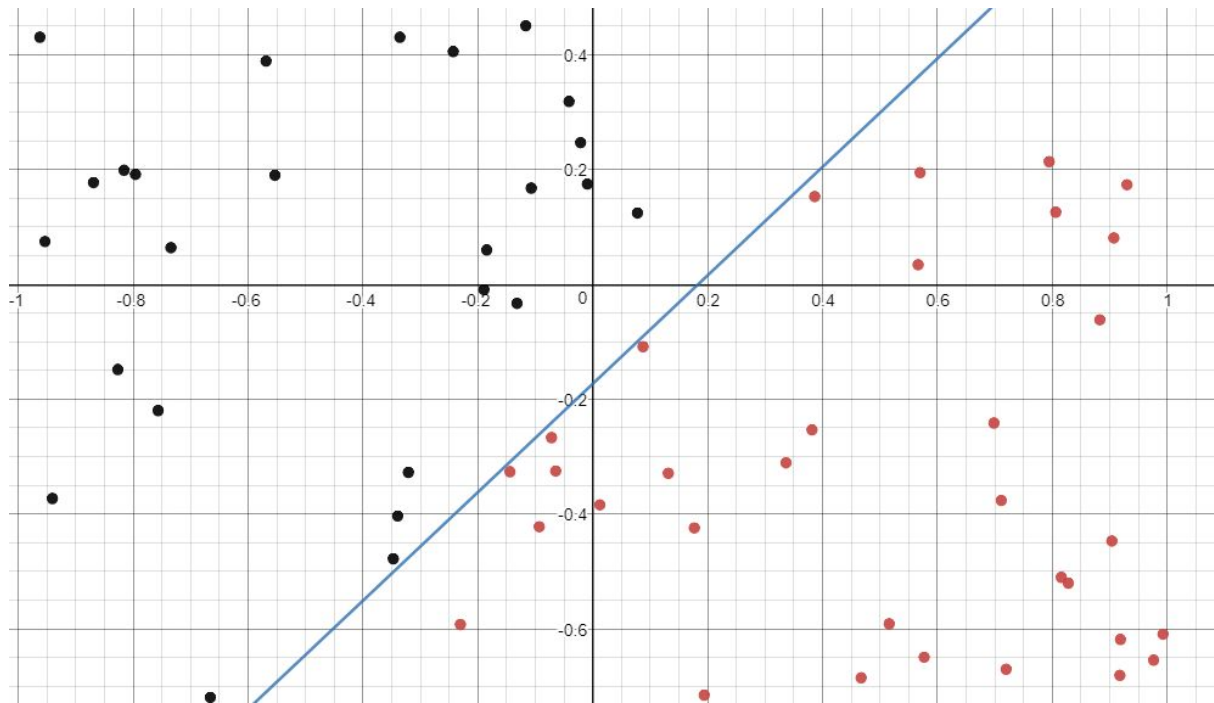
Due Date:

November 14, 2018

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.





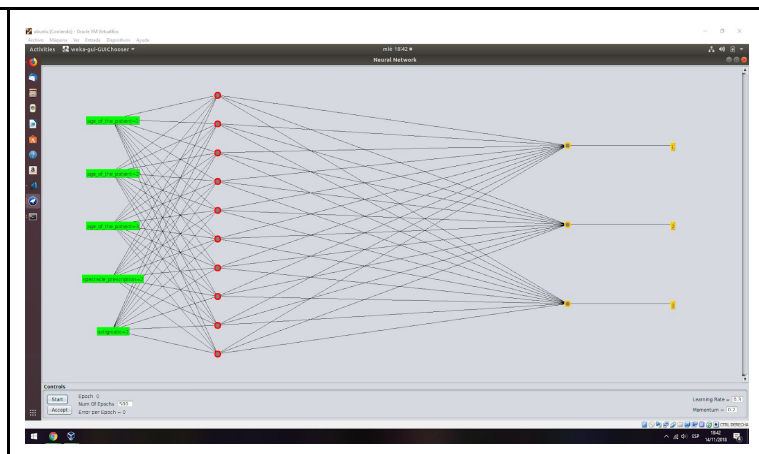
Equation of the line:

$$f(x) = \frac{(1.23813445 \cdot x - 0.22726467547)}{1.3122534872}$$

Part 2

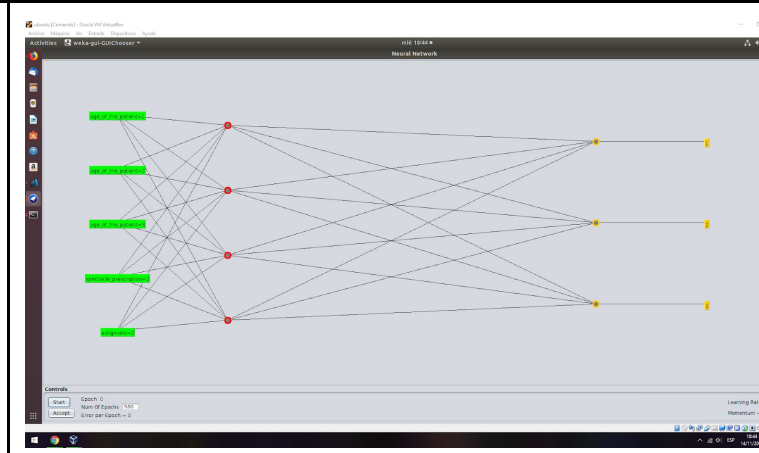
Write an analysis of training the ANN with different parameters. Include an image of the network, the parameters used, and the error and time it took to train each network. Write a brief reflection about what you think is happening in the different ANNs, and why the ANN is behaving like it is. How did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

| | |
|------------------------------|-------------------------------|
| 1- number of hidden layers 4 | 2- number of hidden layers 10 |
|------------------------------|-------------------------------|



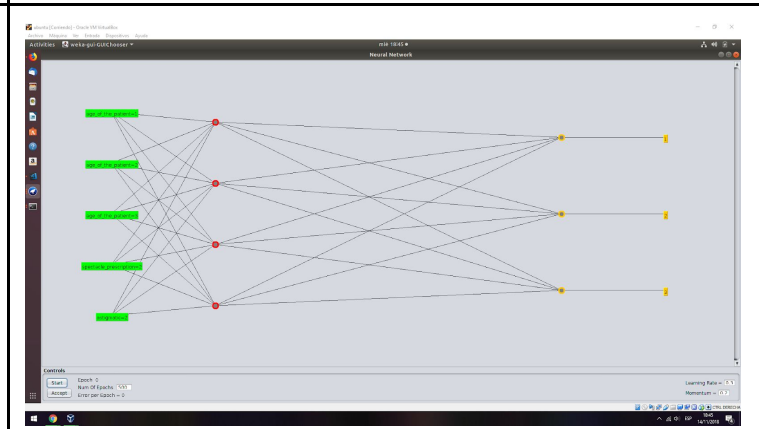
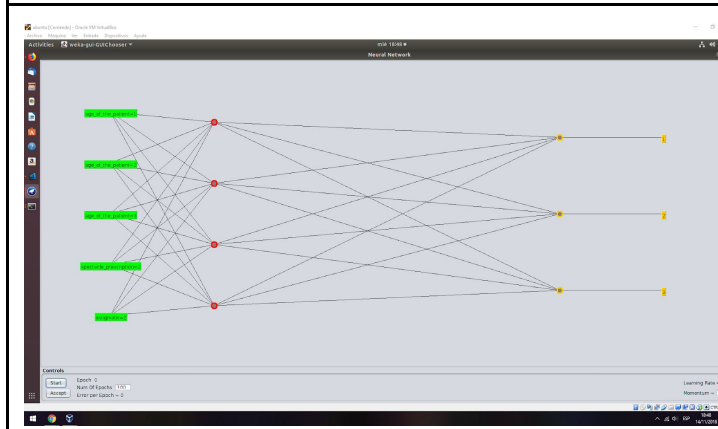
2 learning rate 0.3

2.1 learning rate 0.8



3 Epochs 100

3.1 Epochs 500



| Test | Learning rate | Epochs | Hidden layers | Training time | Correct classification | Incorrect classification |
|------|---------------|--------|---------------|---------------|------------------------|--------------------------|
|------|---------------|--------|---------------|---------------|------------------------|--------------------------|

| | | | | | | |
|-----------------------------------|-----|-----|----|-----|----|----|
| 1 number of hidden layers | 0.3 | 100 | 4 | 100 | 12 | 12 |
| 2 learning rate | 0.3 | 100 | 4 | 100 | 12 | 12 |
| 3 Epochs | 0.3 | 100 | 4 | 100 | 12 | 12 |
| 1.1 number of hidden layers | 0.3 | 100 | 10 | 100 | 12 | 12 |
| 1.2 learning rate | 0.8 | 100 | 4 | 100 | 12 | 12 |
| 1.3 Epochs | 0.3 | 500 | 4 | 100 | 12 | 12 |

We start changing the number of layers, learning rate and epochs, we thought that the changes will be considerable but because the ANN is too small the changes are almost none, maybe if we change the parameters extremely it will present a considerable change. We thought by changing the learning rate to a high value the correct classification should go up, but this was not the case, neither with more epochs and number of hidden layers because of the size of the ANN

Other interesting aspects you could include in your reflection are:

- Explanations as to what are ANNs good for.
 - ANN is very good at solving problems that have linearly separable data. For example for recognitions tests. ANN with Backpropagation is widely used in solving classification and forecasting problems also.
- Where would you use them?
 - It can be used in the recognition of images, like for example the cameras that detect the id numbers of the cars. Using a CNN (convolutional neural network) that this ANN is used for more easily detection and management of patterns.
- Are they worth the effort implementing or not?
 - Depends on the problem, it's important not to start seeing ANN as the golden hammer (to think with AI resolve any problem in the best way possible), because in some cases it isn't. If you got a really simple problem that does not require ANN then if you implement them the effort is not worth it. But, if you got a complex problem like the image recognition that we mention in the last question then the answer is yes it's totally worth it to implement ANN.
- What kinds of problems do they not solve?
 - Any kind of problem that has is not linearly separable.