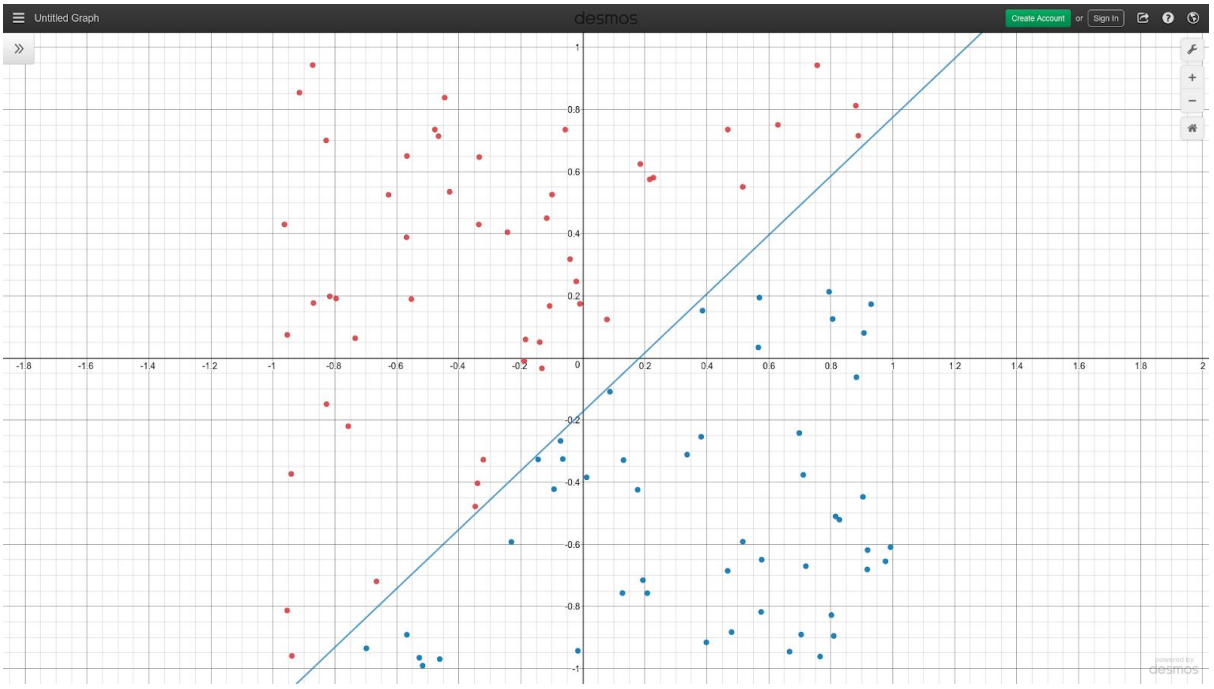
Moises Montaño Copca - A01271656
Diego Alfredo Ballesteros Bautista - A01271588

Lab: Perceptron / Neural Networks

Part 1



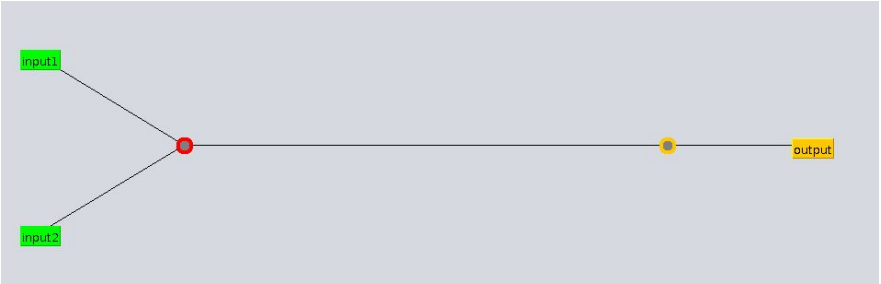
Red = 0
Blue = 1

f(x) = (1.243813445042359x -0.22726467657901972) / 1.3122534872693066

Part 2

"*linearly_separable*" test used.

Original



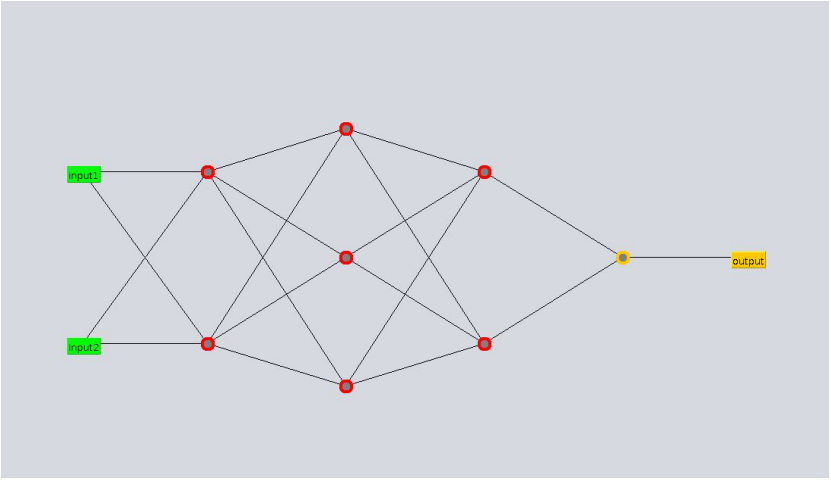
Hidden layers = a

Learning rate = 0.3

Time taken to build model: 0.08 seconds

Time taken to test model on supplied test set: 0.02 seconds

Example 1:



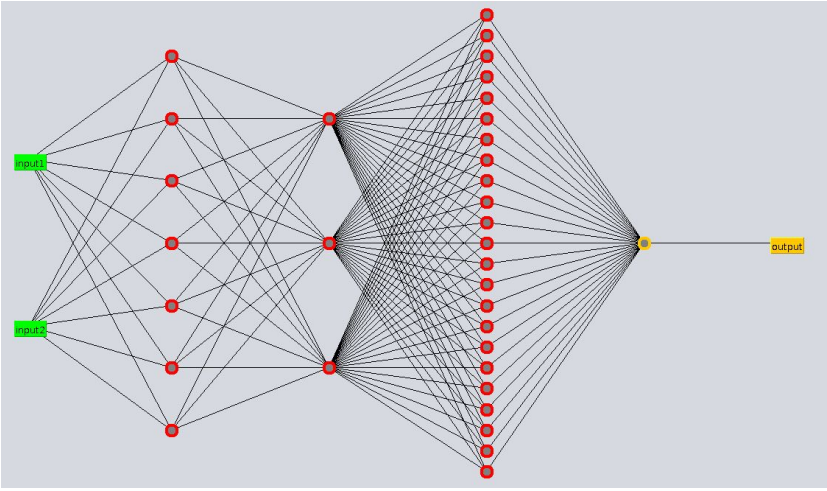
Hidden layers = 2, 3, 2

Learning rate = 0.5

Time taken to build model: 0.1 seconds

Time taken to test model on supplied test set: 0.05 seconds

Example 2:



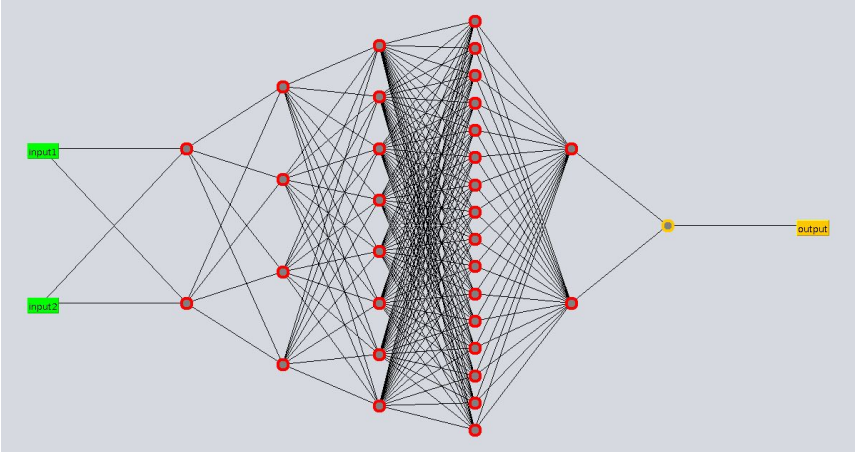
Hidden layers = 7, 3, 27

Learning rate = 0.5

Time taken to build model: 0.27 seconds

Time taken to test model on supplied test set: 0.06 seconds

Example 3:



Hidden layers = 2, 4, 8, 16, 2

Learning rate = 0.2

Time taken to build model: 0.42 seconds

Time taken to test model on supplied test set: 0.04 seconds

While playing with the attributes, adding hidden layers and editing the learning rate, we can see that there are big changes, you can see it with the original and the first example where it takes more than twice in time to test the model with just a few hidden layers added. But incredibly if you lower the learning rate like in the last example (#3), you can see that it's not a big difference in the time taken to test the model, and having a bigger amount of hidden layers. Making us think that the learning rate is an important thing which determines how the neural network will behave with the data.

Other interesting aspects you could include in your reflection are:

- Explanations as to what are ANNs good for.

ANNs are good for too many things, in the medicine area like neuroscience, or predictions of any diseases, also used for image recognition like pattern matching, sequence recognition and face recognition where you have the ANN to identify just faces and classifying it in different classes where you just have the object you want and the images that are not similar to what the ANN is trained for. Also useful to solve problems like predictions, random functions for approximations, and also used for data analysis and data mining.

- Where would you use them?

You can use them for robotic vision for a lot of applications, you can add a neural network to a drone and make it able to recognize any object you want and be able to navigate in any environment. Also it might be perfect to apply ANN in predictions of cryptocurrency exchange rates.

- Are they worth the effort implementing or not?

Yes, as told before it is used for a widely range of areas not just in IT stuff, so if the ANN is perfectly trained, it might save lives in the medical area, being very accurate to predict a disease. We think that it totally worth the effort to learn and start implementing it in real life stuff to make things easier and better.

- What kinds of problems do they not solve?

Its difficult to know the information given from the network structure, and the ANN is unable to read the unlabeled input data or data that is not linearly separable.