INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

Artificial Intelligence

Report Lab Artificial Neural Networks

Equipo:

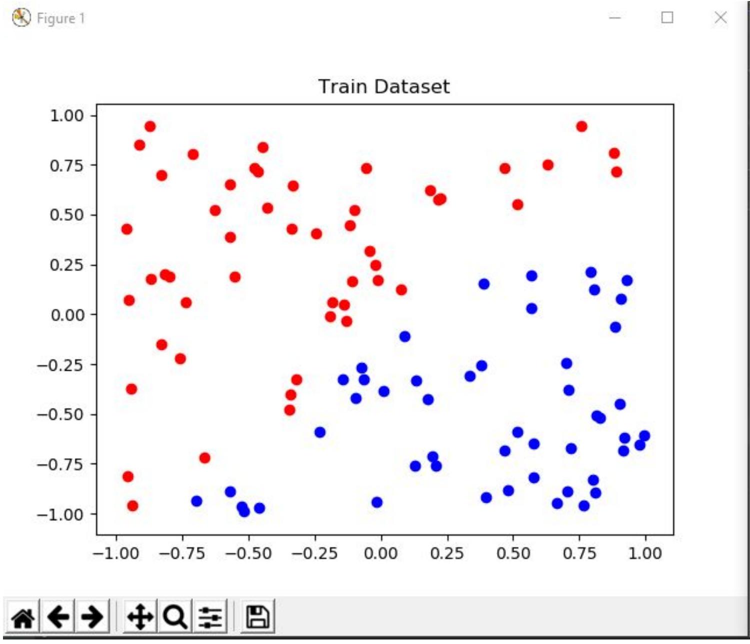
Melannie Isabel Torres Soto A01361808

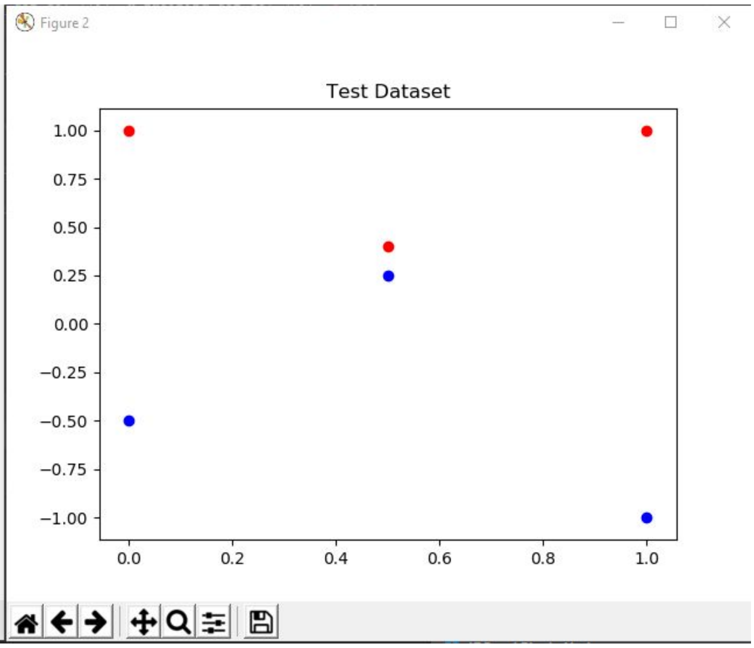
Bernardo Gómez Romero A01209704

Due Date:

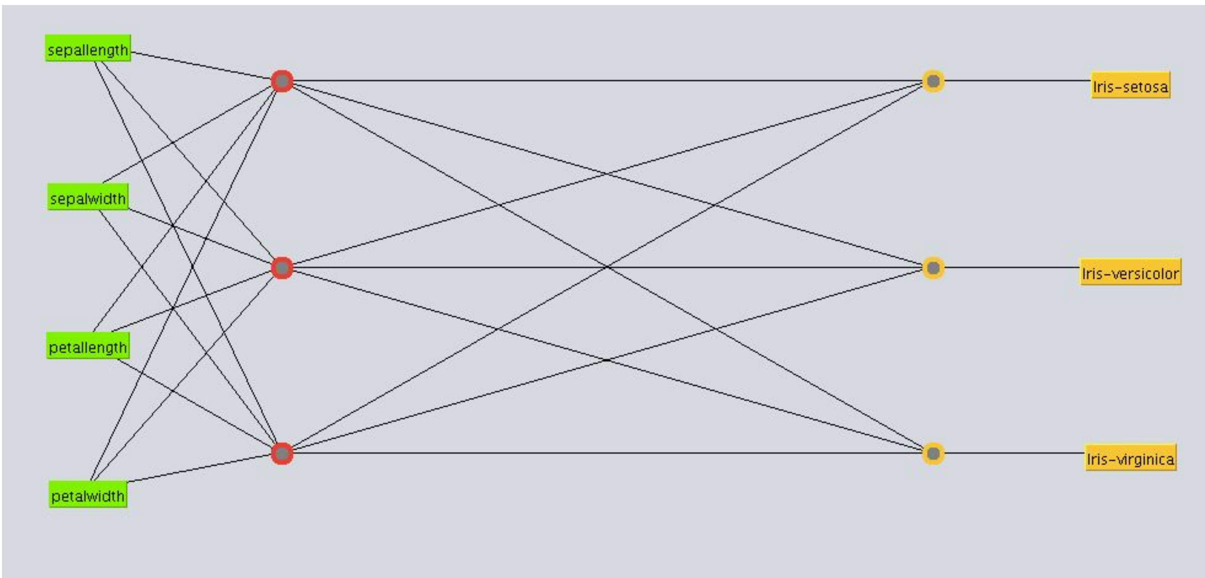
November 14, 2018

Part 1





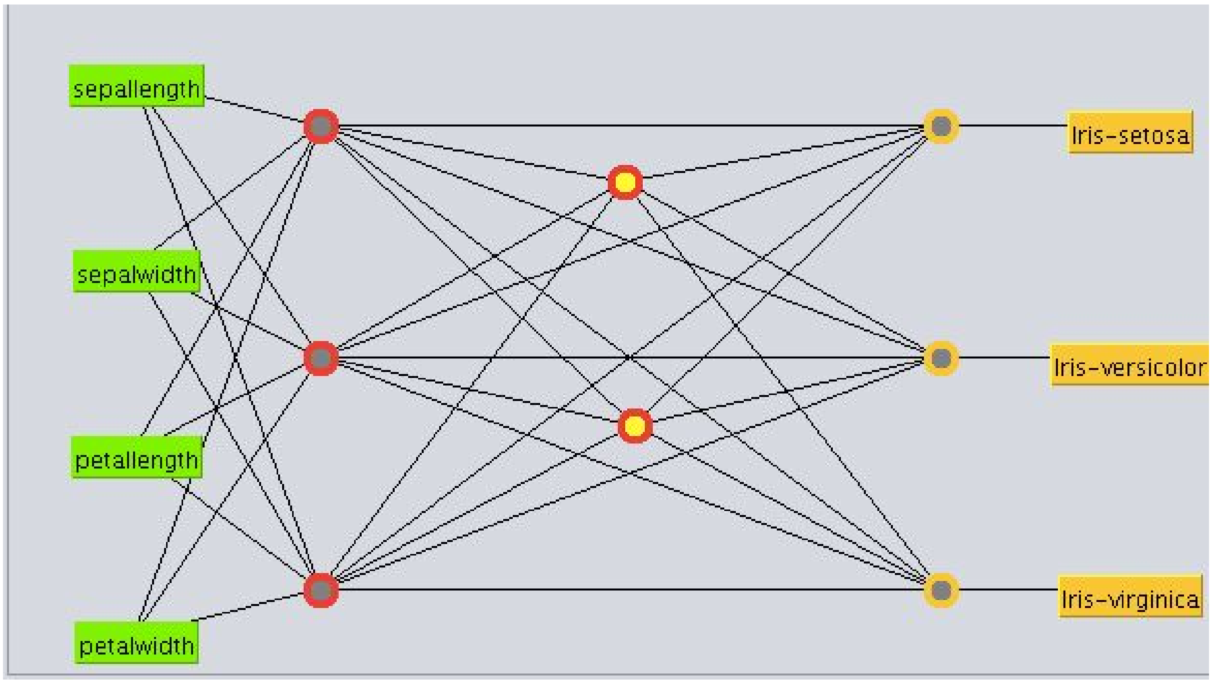
Part 2



Weka Test 1:
Epoch 500
Num Of Epochs 500
Error per Epoch 0.009275
Learning Rate 0.3
3 layers

Weka Test 2:
Epoch 500
Num Of Epochs 500
Error per Epoch 0.0104073
Learning Rate 0.6
3 layers

For test 1 and test 2 we used the same ANN structure: 3 layers (1 input, 1 hidden and 1 output layer), 3 neurons for the hidden layer and 3 neurons for the output layer. The difference between this tests was the learning rate: for test 1 the rate was 0.3 and for test 2, 0.6. The difference we observed in the results was that for a bigger learning rate, the error increased slightly, from 0.009275 to 0.0104073. This is not a very big difference but what it means is that when the learning rate is bigger it can miss the lowest error because the step in the gradient descent is bigger so it just passes over the minimum without detecting it. Test 1, between all our tests was the one with the lowest error.



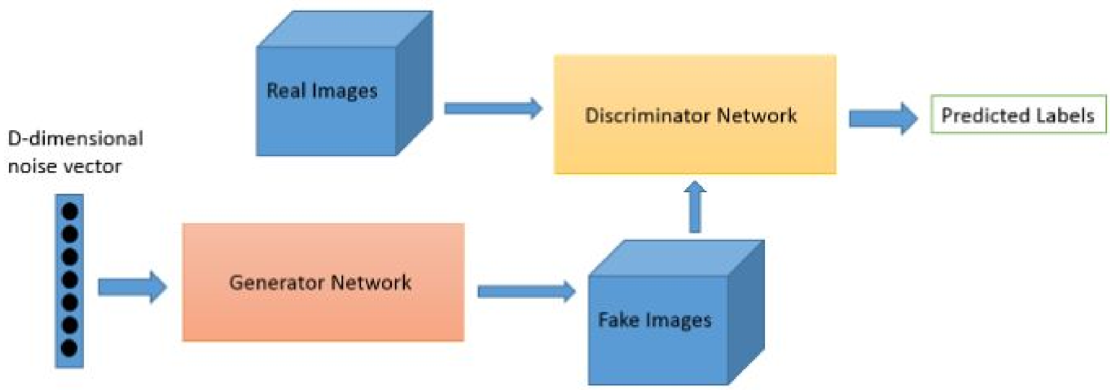
Weka Test 3:
Epoch 500
Num Of Epochs 500
Error per Epoch 0.0094852
Learning Rate 0.3
4 layers

Weka Test 4:
Epoch 500
Num Of Epochs 500
Error per Epoch 0.0095179
Learning Rate 0.6
4 layers

For test 3 and test 4 we used the same ANN structure: 4 layers (1 input, 2 hidden and 1 output layer), with 3 and 2 neurons in the hidden layer and 3 neurons for the output layer. The difference between this tests once more was the learning rate: for test 1 the rate was 0.3 and for test 2, 0.6. Now the difference in the error was smaller, and again the one with the smallest learning rate had the smallest error. One thing worth noting is that in this ANN structure we connected both hidden layers to the output layer. The reason we did this is because when we tried training a network just connecting each layer to its subsequent layer, the program crashed and didn't give us any feedback. With trial and error we notice that this way the network did work so we stick to this approach.

Conclusions

The type of learning ANN were initially intended was for supervised learning since the data used to train the model needs to be labeled. But since the ANNs had earn traction in the market there had been many variations and architectures that expands their capabilities into different applications. One example of this new variations are Generative Adversarial Networks (GANs) which consist in 2 networks competing with each other: one is a clasificator network mapping features to output labels and the generative models the distribution of individual classes (how likely is each class given some input). The way they compete is that one neural network (the generator) generates new data, while the other (the discriminator) evaluates them for authenticity with the following architecture:

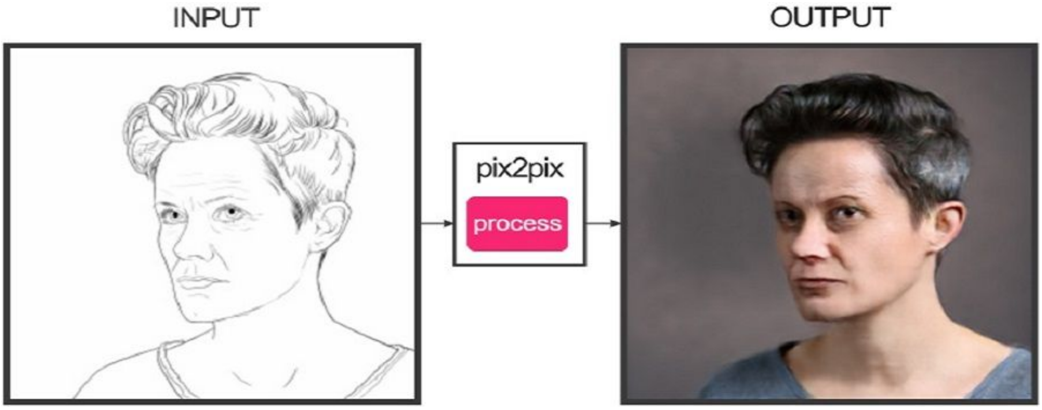


I find this GANs particularly interesting because of their potential to learn from any type of data. in domains that were previously thought to be only possible to dominate by the human brain like: images, music, speech, etc. [2]

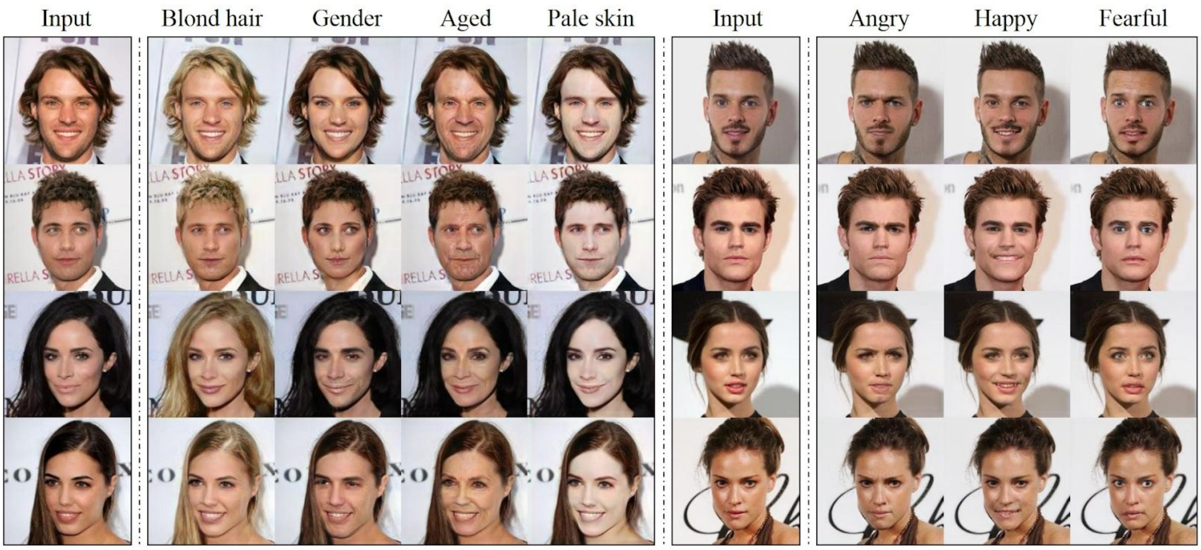
Some examples of its applications:

- mimicking artistic styles:

- sketch to image (pix2pix)



- Face modification:



Sources:

[1]
https://www.tutorialspoint.com/artificial_neural_network/artificial_neural_network_unsupervised_learning.htm
[2] https://skymind.ai/wiki/generative-adversarial-network-gan