Arturo Gamboa A01749802 Hw07, computational intelligence

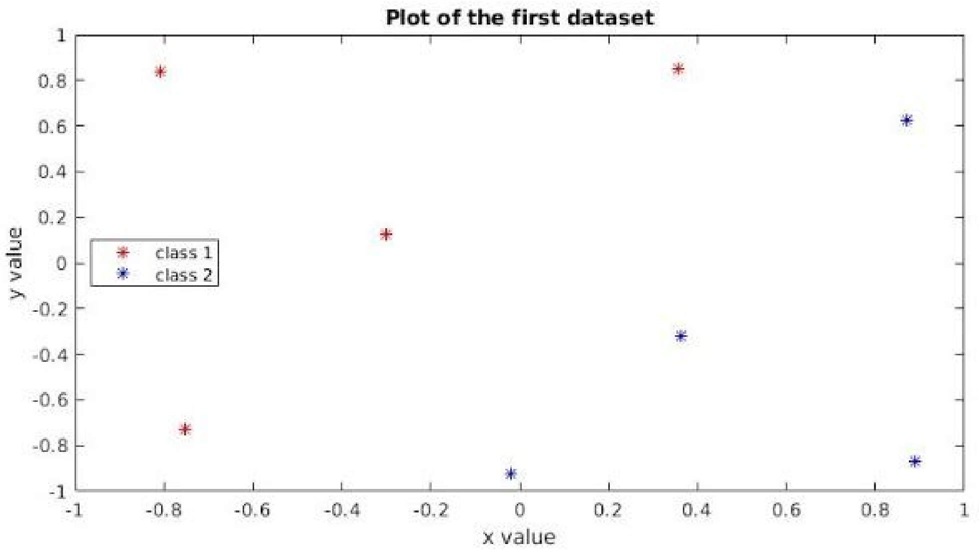
The perceptron’s learning rule in practice

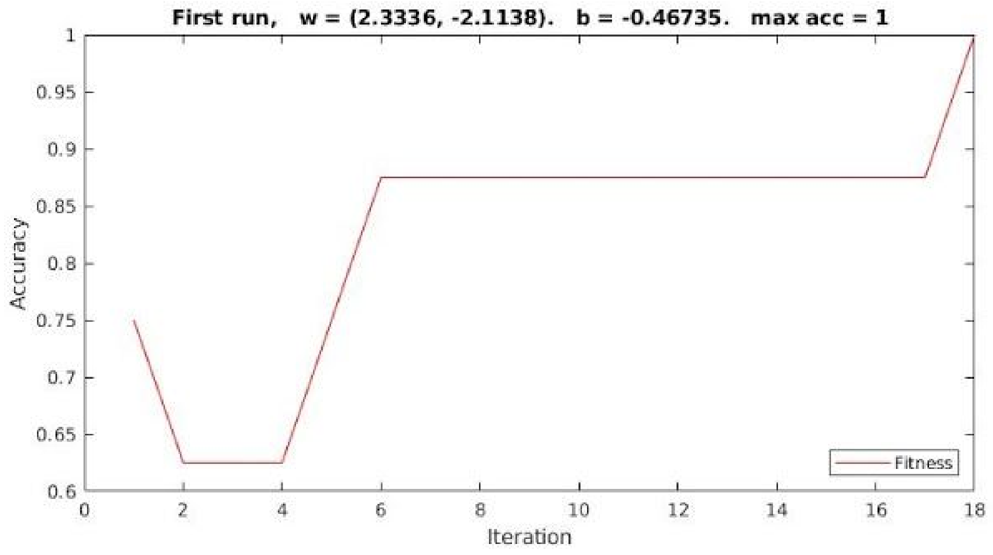
I attach a nice script I made that plots:

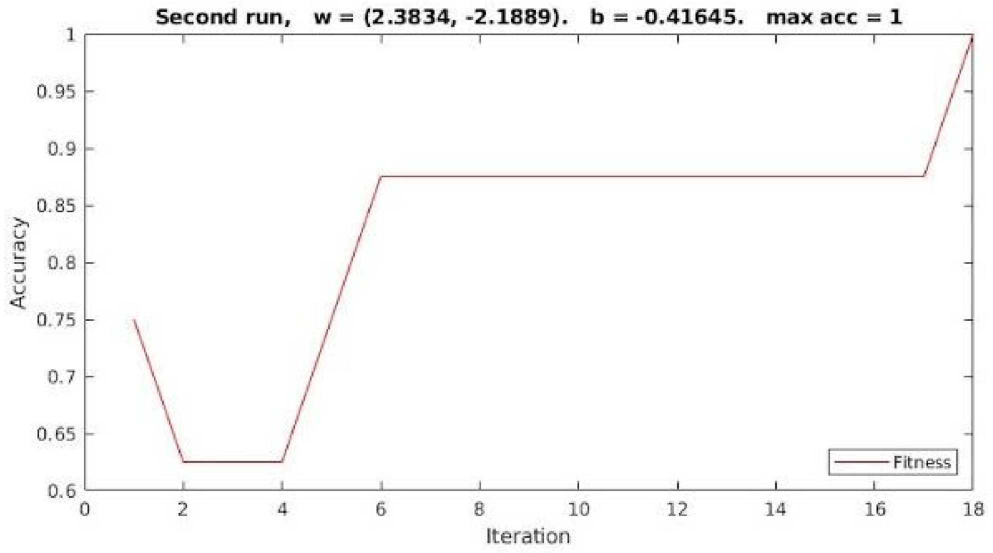
- Three runs for each dataset
- The maximum accuracy of the neuron
- Values of the weights and bias.

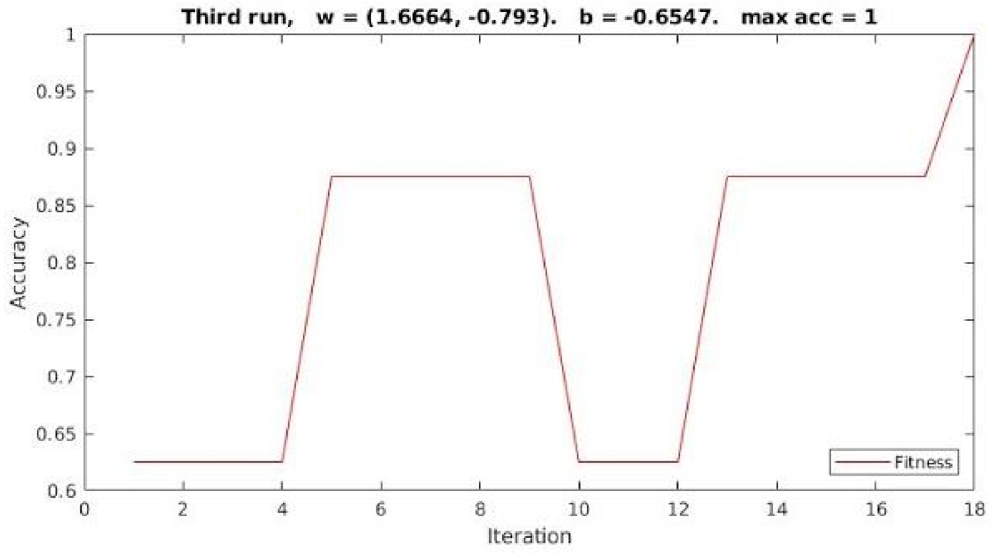
Arturo Gamboa A01749802 Hw07, computational intelligence

The first dataset is linearly separable, so we can use a neuron to classify this data.



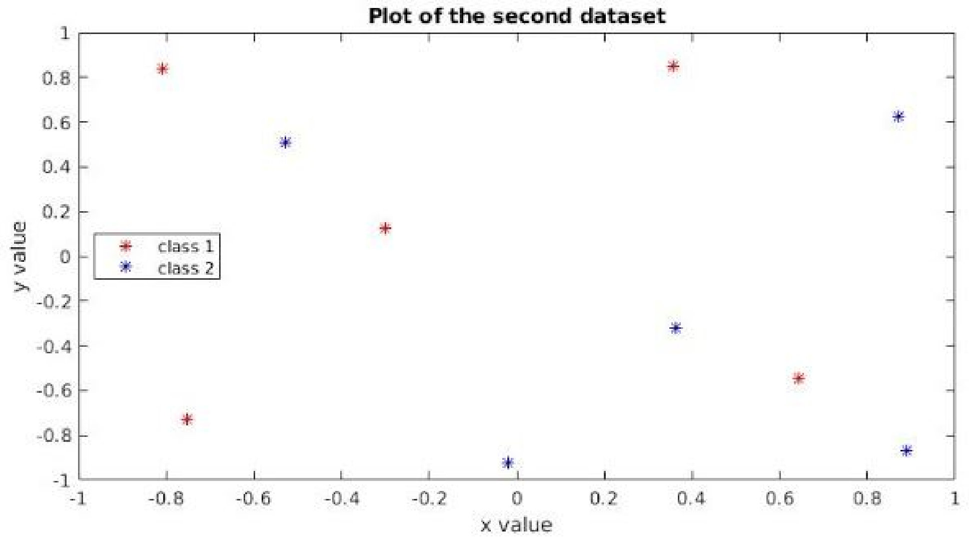


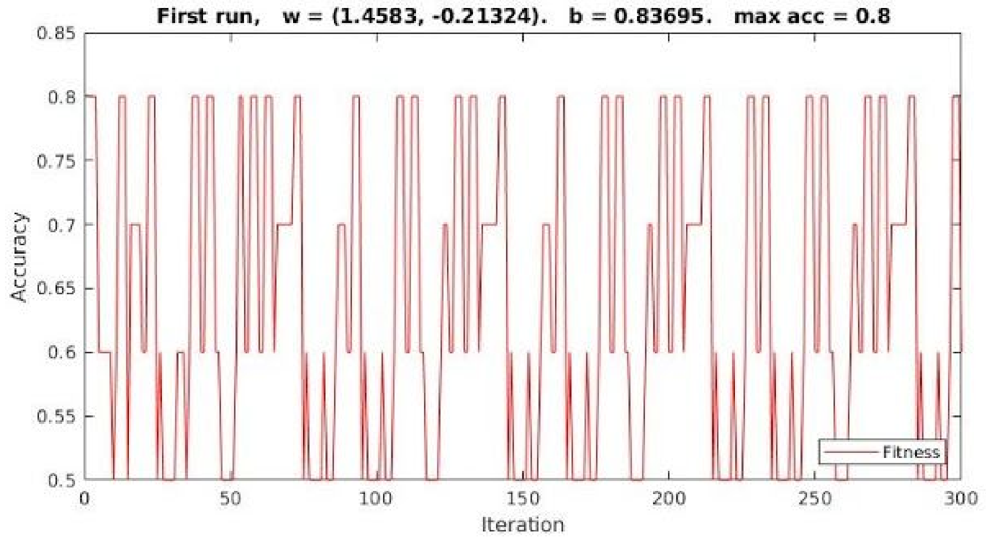


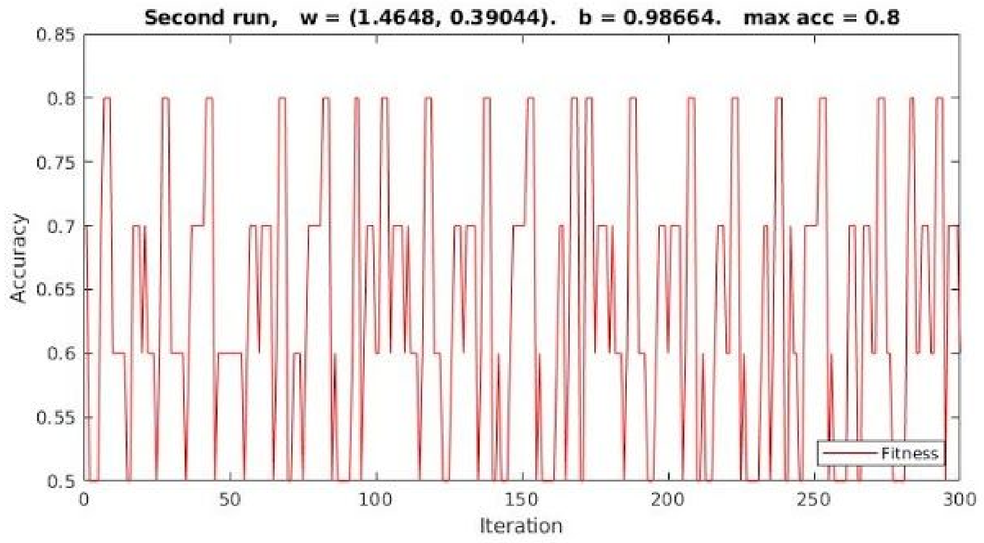


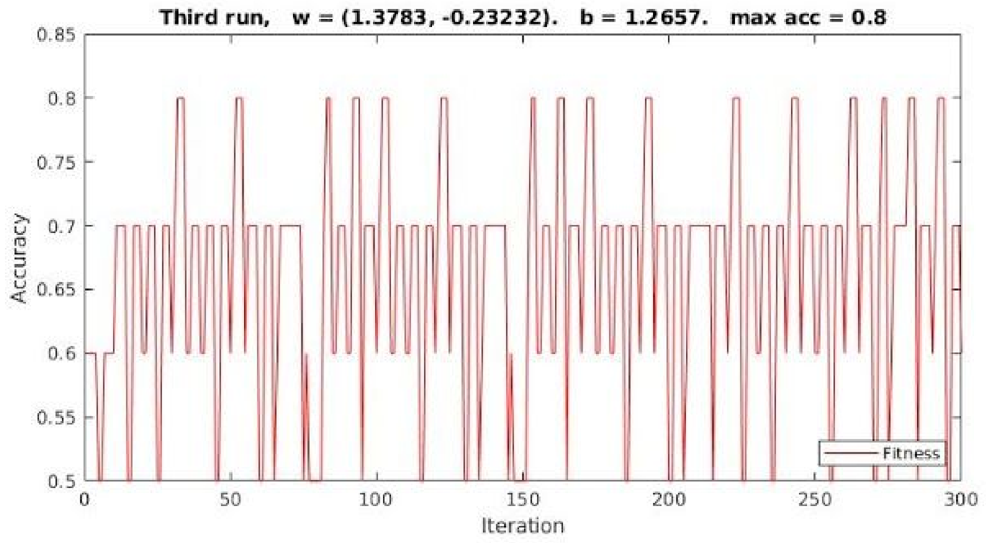
Arturo Gamboa A01749802 Hw07, computational intelligence

The second dataset is not linearly separable, so we can not use a single neuron to classify this data.



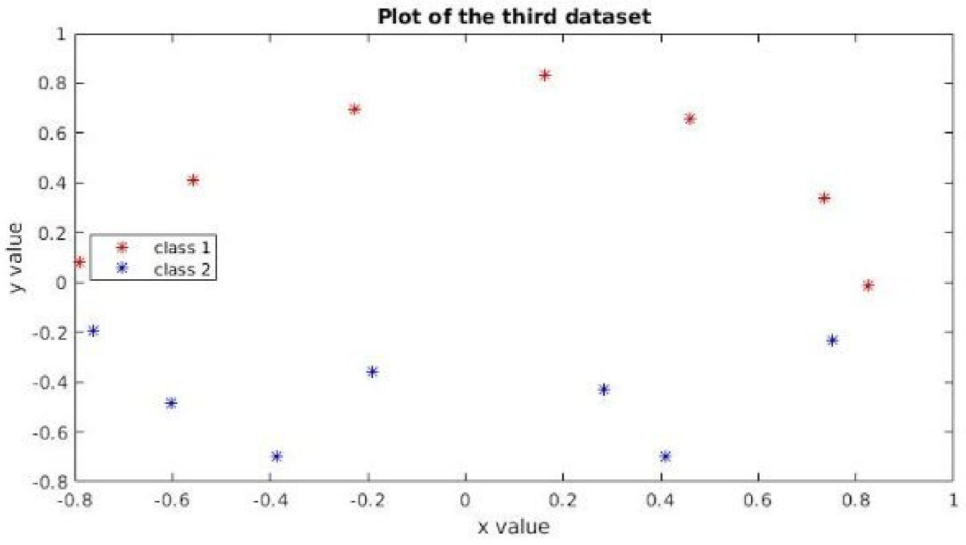


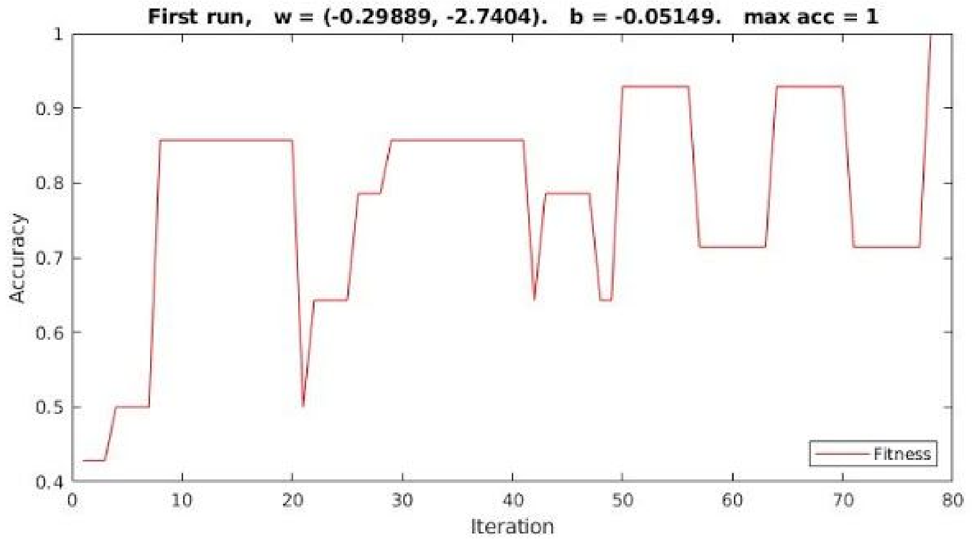


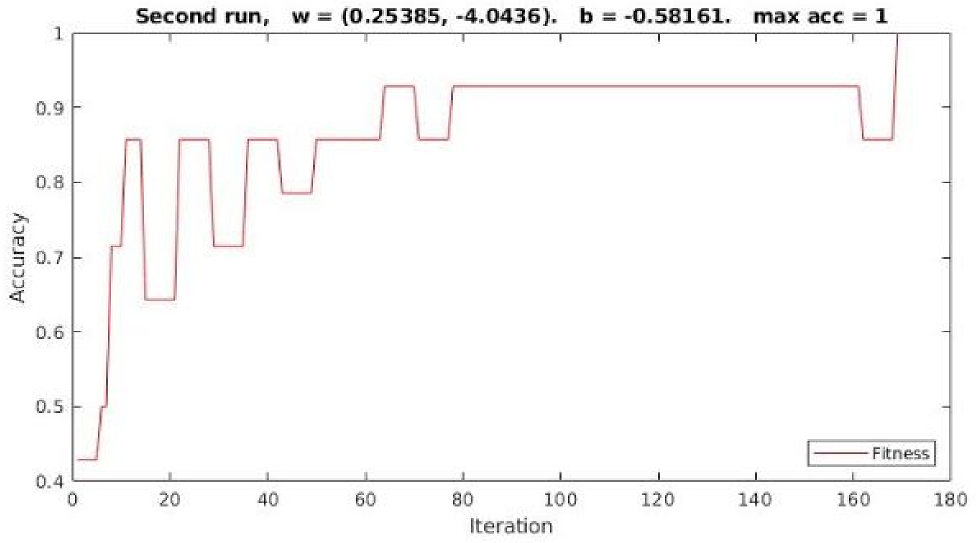


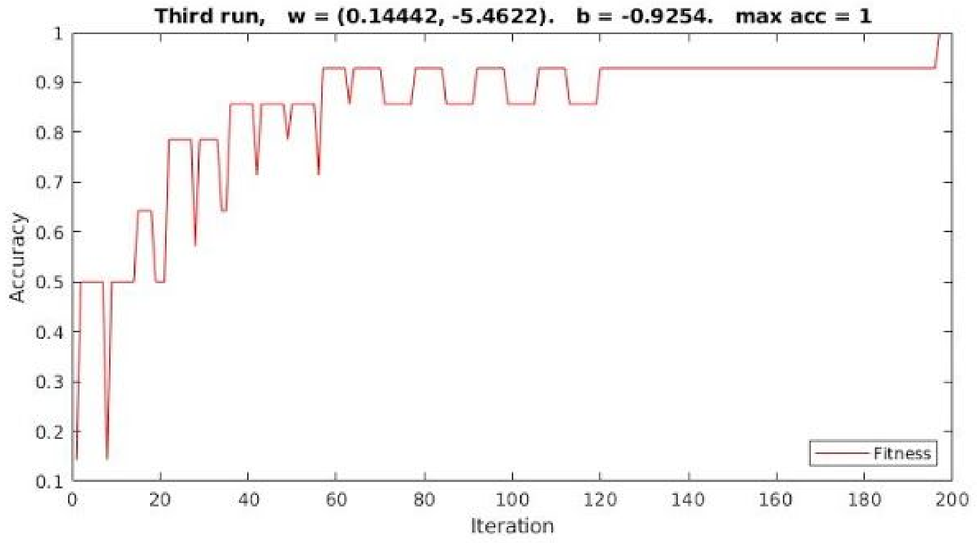
Arturo Gamboa A01749802 Hw07, computational intelligence

The third dataset is linearly separable, and we could find the neuron that classifies the dataset correctly, but it took more time than Datset 1.



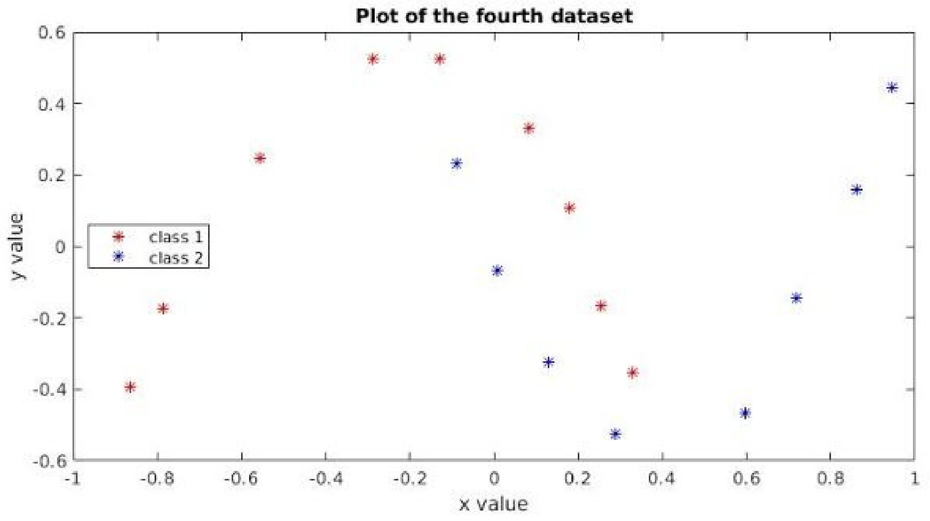


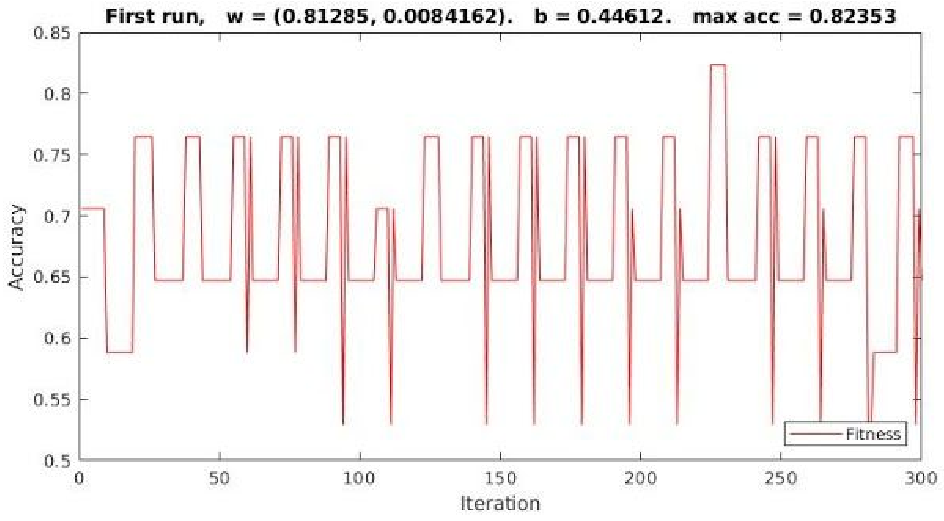


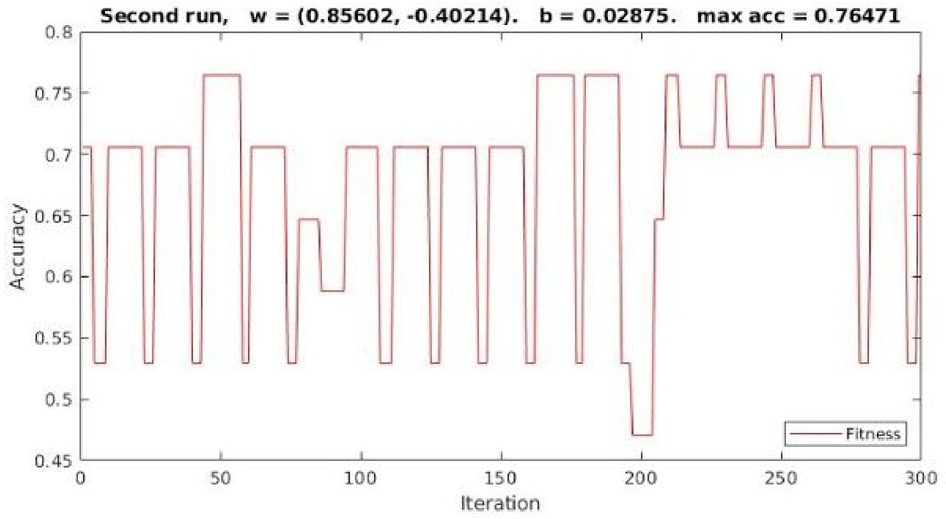


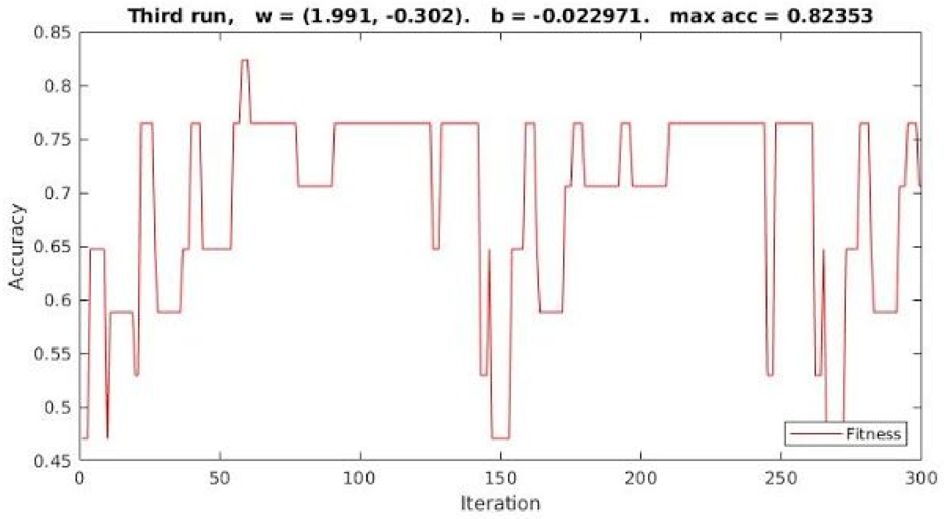
Arturo Gamboa A01749802 Hw07, computational intelligence

The fourth dataset is not linearly separable, so we can not use a single neuron to classify this data.

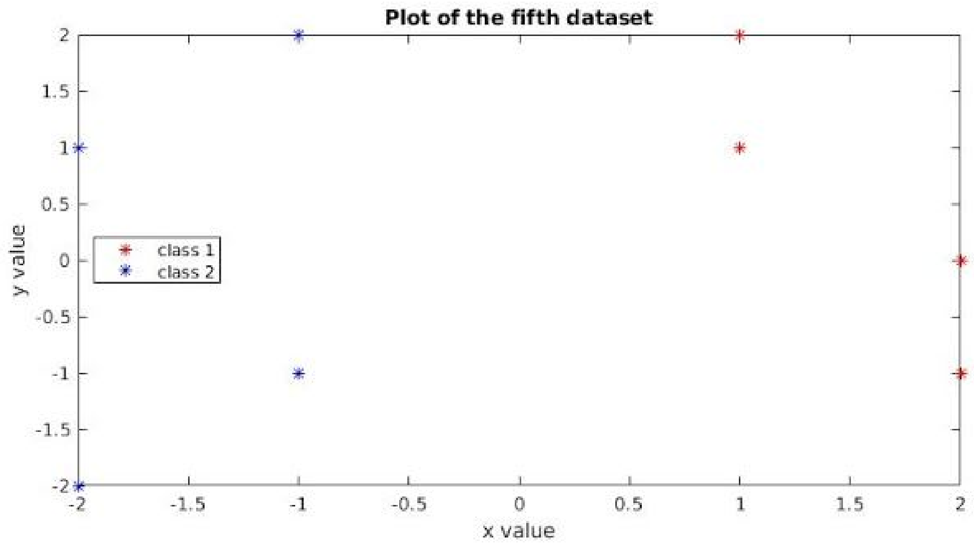


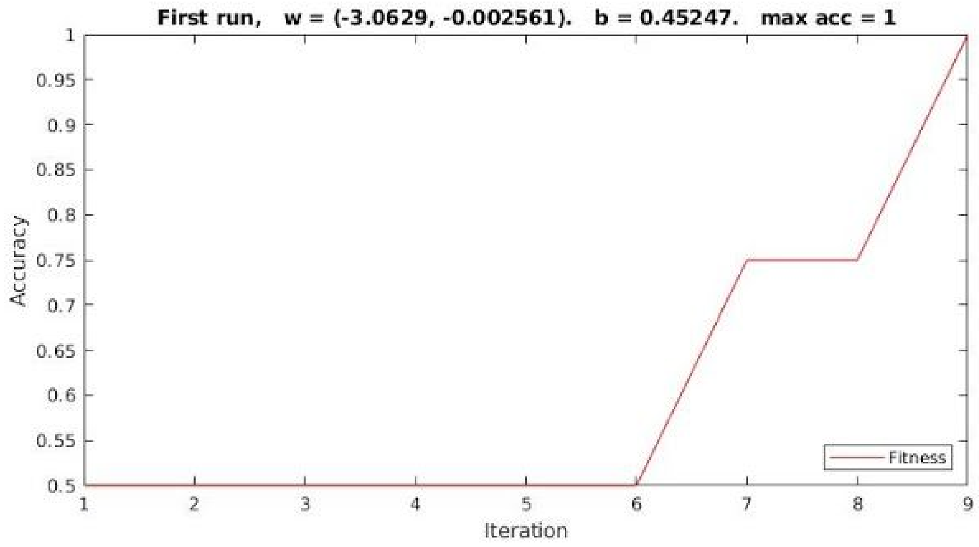


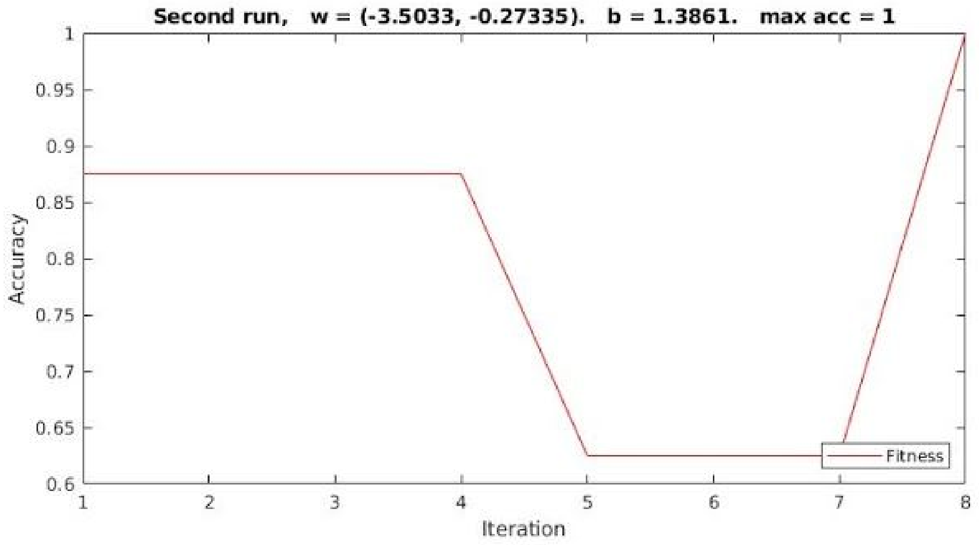


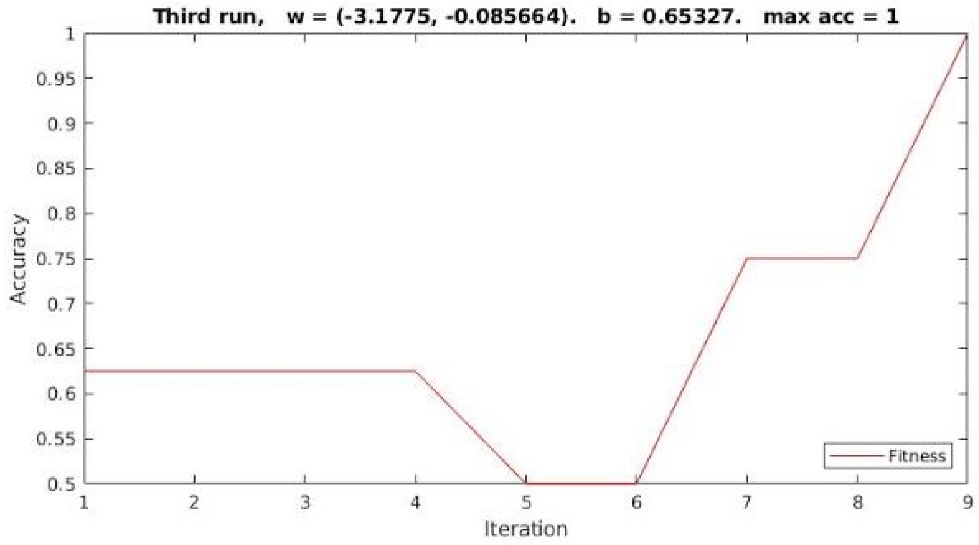


The fifth dataset is easily separable



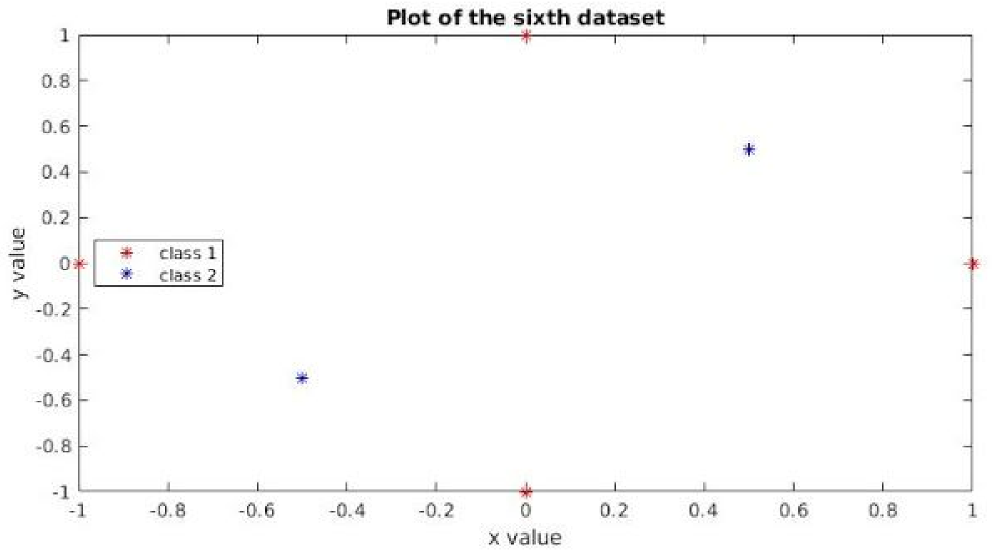


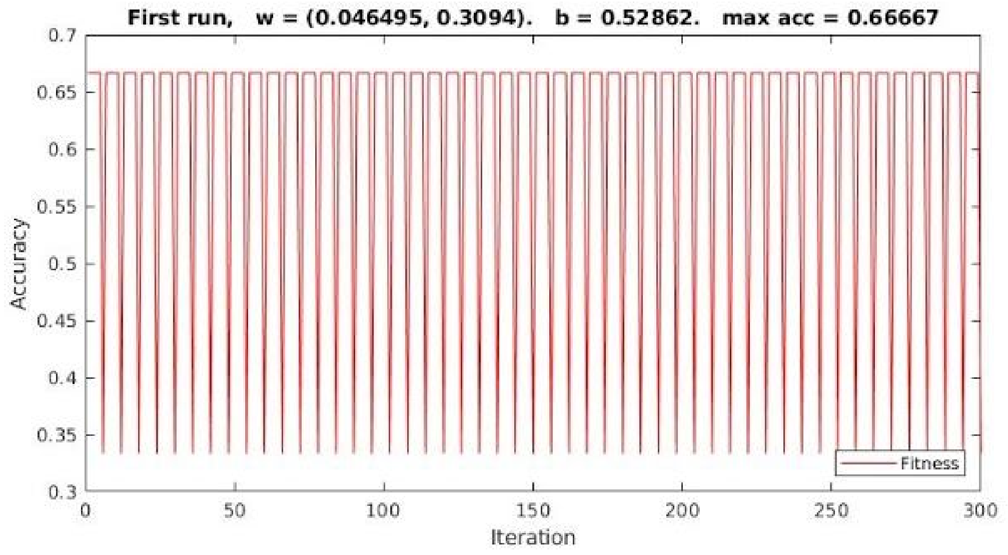


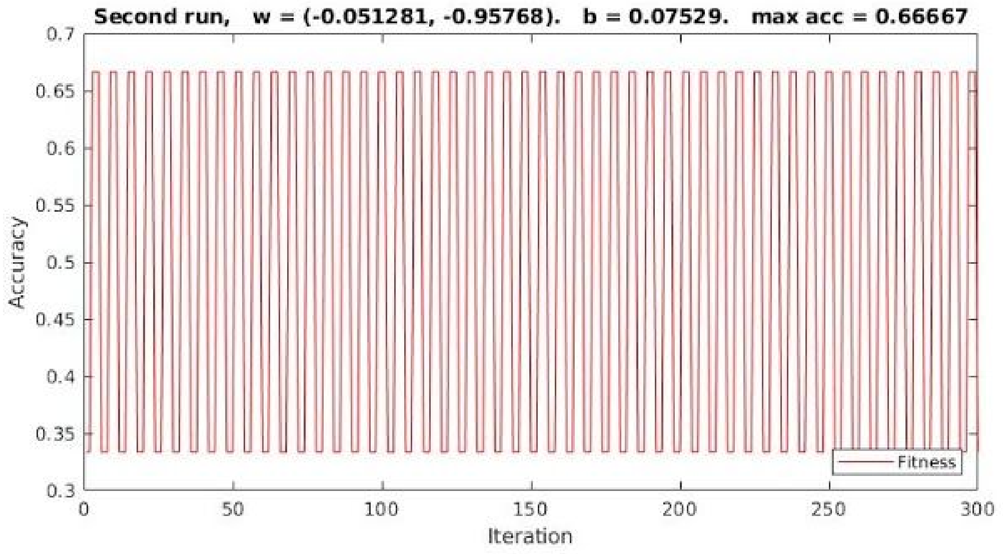


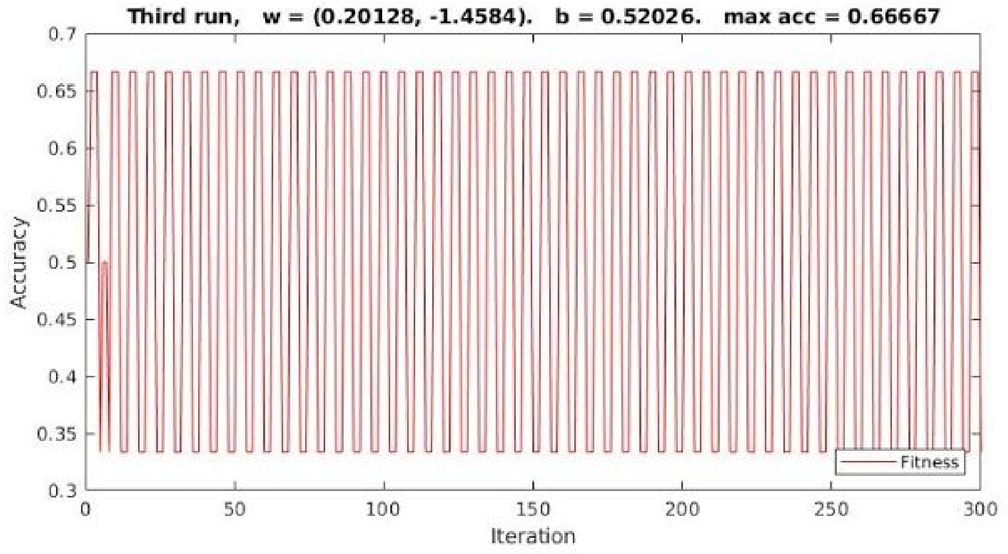
Arturo Gamboa A01749802 Hw07, computational intelligence

The sixth dataset is not linearly separable.





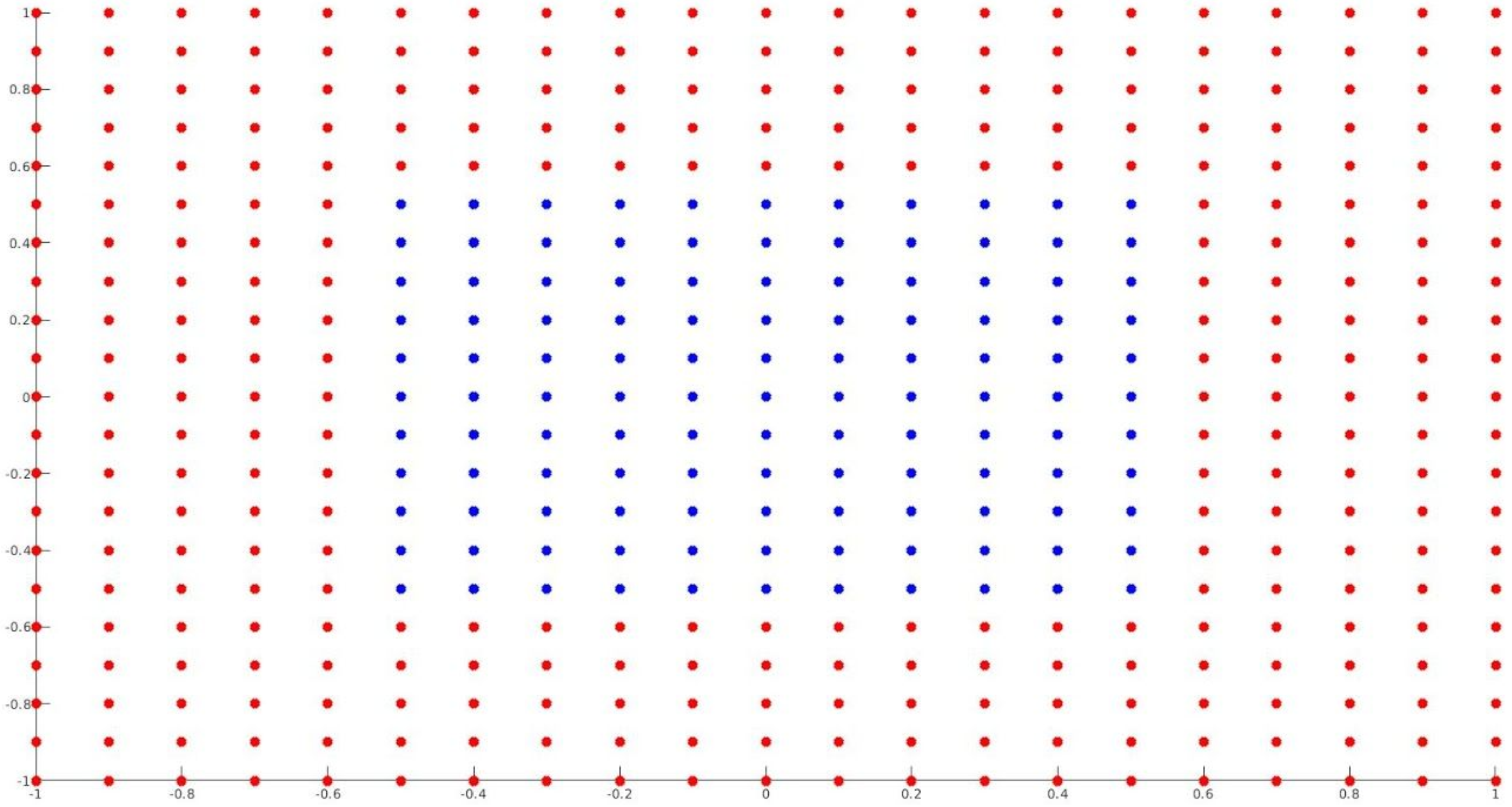




Arturo Gamboa A01749802 Hw07, computational intelligence

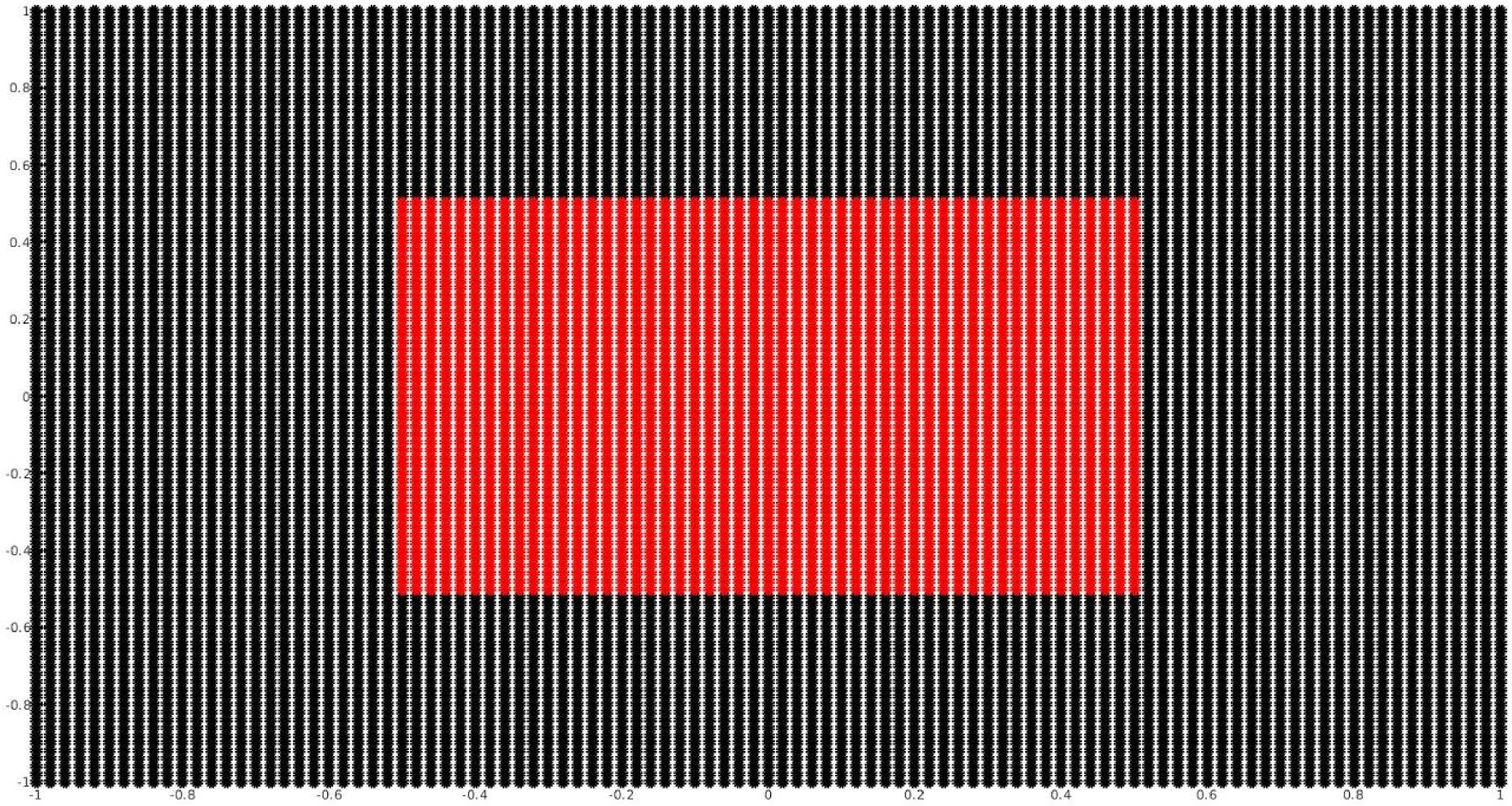
Multi Layered Neural networks

I built a deep neural network that behaves as follows:



Arturo Gamboa A01749802 Hw07, computational intelligence

There is also a higher resolution image of this deep neural network.



The first layer contains neurons that determine the following

- Neuron 1: If the element of the x axis is greater than 0.5 (W = [0, 1] b = -0.51)
- Neuron 2: If the element of the y axis is greater than 0.5 (W = [1, 0] b = -0.51)
- Neuron 3: If the element of the x axis is less than -0.5 (W = [0, -1] b = -0.51)
- Neuron 4: If the element of the x axis is less than -0.5 (W = [-1, 0] b = -0.51)

The second layer contains a single neuron that has as entries the outputs of the first layer, and if a single one of the entries is 1, then the output will be one as well:
Neuron 1: (W = [1, 1, 1, 1], b = -0.1)

All of these code usted here is available in https://github.com/iamerroralpha/Hw07ComputationalIntelligence.

To run the code that corresponds to *the perceptron's learning rule in practice,* just clone the repo, and then run ***cod1.m***.
To run the code that corresponds to *multi layered neural networks*, clone the repo and run ***deepscript.m***.