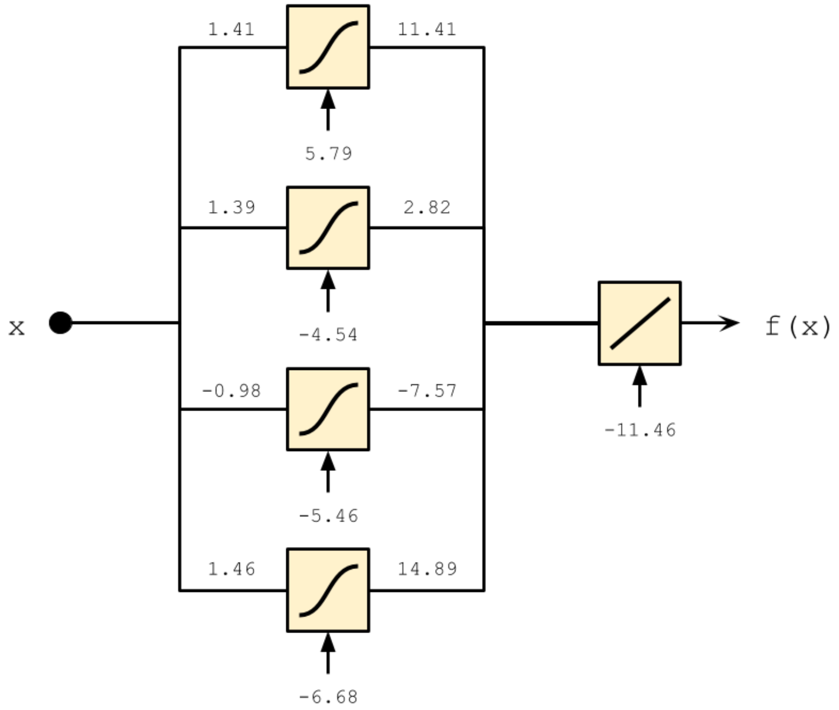
Evolutionary Computation (CS5048)

Challenge 01

In this challenge, you will use a genetic algorithm to adjust the weights of a simple neural network. If properly set, the weights in the neural network will make it possible to approximate a particular function (unknown to you). The value of this challenge is 2% additional to the final grade (applicable only on modules 01 and 04). This challenge can be solved in teams of, at most, four people.

1 The neural network

The neural network that will be used in this homework contains two layers of neurons. The first layer contains neurons with a log-sigmoid activation function ($\frac{1}{1+e^{-n}}$) while the output layer contains neurons with a linear activation function ($a = n$). Since the function to approximate has only one independent variable (x), the only input is connected to all the neurons in the first layer. The number of neurons in the first layer (the neurons with a log-sigmoid activation function) may vary, based on the configuration defined by the user. For example, a neural network with four neurons in the first layer may look like as follows:

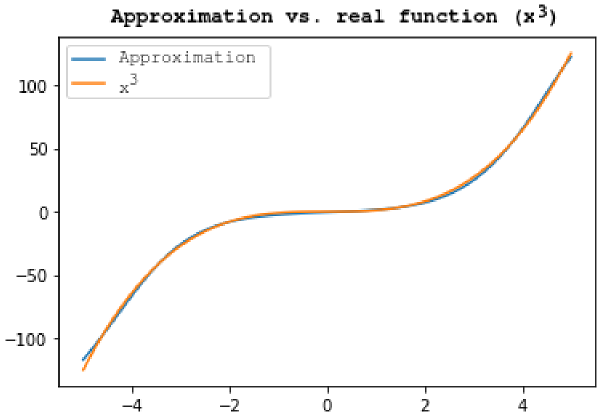


This network approximates the function $f(x) = x^3$, in the range $[-5, 5]$, with a sum of squared error (SSE) between the approximated function and the real one of 2.92. The SSE is calculated as $SSE = \sum_{i=1}^{n}(x_i - x_i^*)^2$, where x_i and x_i^* are the results obtained for case i by the approximated function and the real one, respectively.

The weights of this neural network can be represented as:

$$W^1 = \begin{bmatrix} 1.41 \\ 1.39 \\ -0.98 \\ 1.46 \end{bmatrix} \qquad W^2 = \begin{bmatrix} 11.41 & 2.82 & -7.57 & 14.89 \end{bmatrix}$$

$$b^1 = \begin{bmatrix} 5.79 \\ -4.54 \\ -5.46 \\ -6.68 \end{bmatrix} \qquad b^2 = \begin{bmatrix} -11.46 \end{bmatrix}$$



2 The genetic algorithm

Implement a genetic algorithm that finds the weights for a neural network so that it approximates an unknown function described by the partial data distributed along with this document (`data.csv`). Although I recommend using a topology similar to the one described in the example, this is something you can choose from many options. Them feel free to use the netowrk architecture that best fits your needs. Your chromosome will use a real-valued representation, and to evaluate it, you must find a way to extract the information from the chromosome to generate the matrices of weights and biases required for the network to operate. For example, one way to represent the previous neural network could be with a chromosome of 13 genes, in the following way:

$$\underbrace{\boxed{1.41}\boxed{1.39}\boxed{-0.98}\boxed{1.46}}_{W^1}\underbrace{\boxed{5.79}\boxed{-4.54}\boxed{-5.46}\boxed{-6.68}}_{b^1}\underbrace{\boxed{11.41}\boxed{2.82}\boxed{-7.57}\boxed{14.89}}_{W^2}\underbrace{\boxed{-11.46}}_{b^2}$$

Of course, the evaluation function within the genetic algorithm must interpret the chromosome in such a way that the values for the weights in the neural network are properly set.

To help you with this homework, I have distributed a sample Python code that approximates a the function $f(x) = x^3$. Feel free to use this code at your best convenience.

Deliverables

Prepare a ZIP file that contains your implementation of the genetic algorithm that approximates the unknown function given the data provided and a `readme` file with the detailed steps to run the codes and a PDF that contains the information about the team, the configuration used to find the best approximation, the SSE value obtained and a plot of such approximation in the range $[-5, 5]$. The PDF file must also describe the network topology and the weights found, so I can replicate your findings. Submit the ZIP file to Canvas. If working as part of a team, only one of the team members should submit the ZIP file. **Please, do not submit other formats but ZIP**. Please consider that failing to submit a valid `readme` file or providing incomplete or wrong instructions may invalidate your submission.

I promise to apply my knowledge, strive for its development, and not use unauthorized or illegal means to complete this activity, following the Tecnológico de Monterrey Student Code of Honor.