README OF DNN CUSTOM EQUATION SYSTEM 1 DATABASES THAT WERE GENERATED

made by: César Miranda Meza

email: cmirandameza3@hotmail.com

The Deep Neural Network (DNN) custom equation that was used as a reference to create the databases labeled as "the DNN custom equation system 1" is given by the following:

$$N_{e(1,1)} = \tanh\left(\omega_{(1,1,0)} + \omega_{(1,1,1)}x_1 + \omega_{(1,1,2)}x_2\right) \tag{1}$$

$$N_{e(2,1)} = \omega_{(2,1,0)} + \omega_{(2,1,1)}x_1 + \omega_{(2,1,2)}x_2 \tag{2}$$

$$N_{e(3,1)} = \frac{1}{1 + e^{\left(\omega_{(3,1,0)} + \omega_{(3,1,1)}x_1 + \omega_{(3,1,2)}x_2\right)}} \tag{3}$$

$$N_{e(4,1)} = \left(\omega_{(4,1,0)} + \omega_{(4,1,1)}x_1 + \omega_{(4,1,2)}x_2\right)^4 \tag{4}$$

$$N_{e(5,1)} = e^{\left(\omega_{(5,1,0)} + \omega_{(5,1,1)}x_1 + \omega_{(5,1,2)}x_2\right)} \tag{5}$$

Where $N_{e(1,1)}$, $N_{e(2,1)}$, $N_{e(3,1)}$, $N_{e(4,1)}$ and $N_{e(5,1)}$ are the neurons and their corresponding activation functions (hyperbolic tangent, 1st order degree, logistic, 4th order degree and first order exponential respectively) defined for the first layer of the DNN model to be defined and where its weight values are:

- $\omega_{(1,1,0)} = 0$
- $\omega_{(1,1,1)} = -0.4$
- $\omega_{(1,1,2)} = 0.4$
- $\omega_{(2,1,0)} = -2.34$
- $\omega_{(2,1,1)} = 0.09$
- $\omega_{(2,1,2)} = 0.77$
- $\omega_{(3,1,0)} = -1.26$
- $\omega_{(3,1,1)} = -0.3$
- $\omega_{(3,1,2)} = 0.5$
- $\omega_{(4,1,0)} = -0.91$
- $\omega_{(4,1,1)} = 0.04$
- $\omega_{(4,1,2)} = -0.0037$
- $\omega_{(5,1,0)} = 1.64$

- $\omega_{(5,1,1)} = -0.022$
- $\omega_{(5,1,2)} = -0.022$

$$N_{e(1,2)} = \omega_{(1,2,0)} + \omega_{(1,2,1)}N_{e(1,1)} + \omega_{(1,2,2)}N_{e(2,1)} + \omega_{(1,2,3)}N_{e(3,1)} + \omega_{(1,2,4)}N_{e(4,1)} + \omega_{(1,2,5)}N_{e(5,1)} \quad (6)$$

$$N_{e(2,2)} = \tanh\begin{pmatrix} \omega_{(2,2,0)} + \omega_{(2,2,1)}N_{e(1,1)} + \omega_{(2,2,2)}N_{e(2,1)} + \\ \omega_{(2,2,3)}N_{e(3,1)} + \omega_{(2,2,4)}N_{e(4,1)} + \omega_{(2,2,5)}N_{e(5,1)} \end{pmatrix} \quad (7)$$

$$N_{e(3,2)} = e^{\left(\omega_{(3,2,0)}+\omega_{(3,2,1)}N_{e(1,1)}+\omega_{(3,2,2)}N_{e(2,1)}+\omega_{(3,2,3)}N_{e(3,1)}+\omega_{(3,2,4)}N_{e(4,1)}+\omega_{(3,2,5)}N_{e(5,1)}\right)^2} \quad (8)$$

$$N_{e(4,2)} = \begin{pmatrix} \omega_{(4,2,0)} + \omega_{(4,2,1)}N_{e(1,1)} + \omega_{(4,2,2)}N_{e(2,1)} + \\ \omega_{(4,2,3)}N_{e(3,1)} + \omega_{(4,2,4)}N_{e(4,1)} + \omega_{(4,2,5)}N_{e(5,1)} \end{pmatrix}^6 \quad (9)$$

Where $N_{e(1,2)}$, $N_{e(2,2)}$, $N_{e(3,2)}$ and $N_{e(4,2)}$ are the neurons and their corresponding activation functions (1st order degree, hyperbolic tangent, 2nd order degree exponential and 6th order degree respectively) defined for the second layer of the DNN model to be defined and where its weight values are:

- $\omega_{(1,2,0)} = 0$
- $\omega_{(1,2,1)} = 1$
- $\omega_{(1,2,2)} = 0.1$
- $\omega_{(1,2,3)} = 1$
- $\omega_{(1,2,4)} = 1$
- $\omega_{(1,2,5)} = 2.19$
- $\omega_{(2,2,0)} = 0$
- $\omega_{(2,2,1)} = 1$
- $\omega_{(2,2,2)} = -0.1$
- $\omega_{(2,2,3)} = 1$
- $\omega_{(2,2,4)} = 0.43$
- $\omega_{(2,2,5)} = 1.38$
- $\omega_{(3,2,0)} = -0.2$
- $\omega_{(3,2,1)} = -1.1$
- $\omega_{(3,2,2)} = 0.01$
- $\omega_{(3,2,3)} = -0.02$
- $\omega_{(3,2,4)} = 0.0074$
- $\omega_{(3,2,5)} = 0.017$
- $\omega_{(4,2,0)} = -0.002$
- $\omega_{(4,2,1)} = 0.02$
- $\omega_{(4,2,2)} = -0.002$
- $\omega_{(4,2,3)} = -0.3$

- $\omega_{(4,2,4)} = -0.019$
- $\omega_{(4,2,5)} = 0.017$

$$N_{e(1,3)} = \frac{1}{1 + e^{\left(\omega_{(1,3,0)} + \omega_{(1,3,1)} N_{e(1,2)} + \omega_{(1,3,2)} N_{e(2,2)} + \omega_{(1,3,3)} N_{e(3,2)} + \omega_{(1,3,4)} N_{e(4,2)}\right)}} \tag{10}$$

Where $N_{e(1,3)}$ is the neuron and its corresponding activation function (logistic) defined for the last/output layer of the DNN model to be defined and where its weight values are:

- $\omega_{(1,3,0)} = 1$
- $\omega_{(1,3,1)} = 0.4$
- $\omega_{(1,3,2)} = -0.9$
- $\omega_{(1,3,3)} = -1$
- $\omega_{(1,3,4)} = -1.27$

For the Eqs. (1) to (10), x_1 and x_2 represent the independent variables (inputs of the current sample); $N_{e(1,3)}$ stands for the dependent variable (output of the current sample); and the databases generated with Eq. (10) were restricted such that $0 < x_1 \leq 100$ and $0 < x_2 \leq 100$. Consequently, the expected graphical output of such equation should be the following:

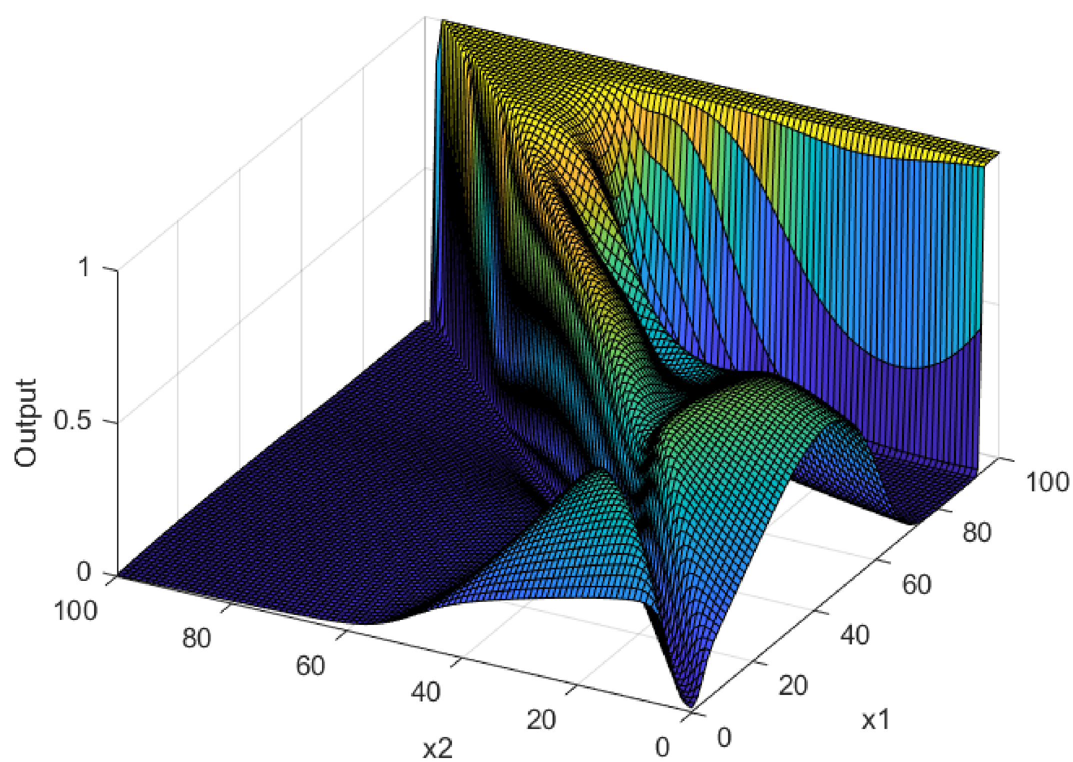

Figure 1-1: Expected 3D graphical output of the Eq. (10).

However, the Eq. (10) was modified by adding to it a bias component r, that would represent a random value and should be generated each time a new sample is calculated:

$$N_{e(1,3)} = \frac{1}{1+e^{\left(\omega_{(1,3,0)}+\omega_{(1,3,1)}N_{e(1,2)}+\omega_{(1,3,2)}N_{e(2,2)}+\omega_{(1,3,3)}N_{e(3,2)}+\omega_{(1,3,4)}N_{e(4,2)}\right)}}+r \quad | -0.1 \leq r \leq 0.1 \quad (11)$$

Where the independent variables were restricted to be sampled with values according to the following way: $0 < x_1 \leq 100$ and $0 < x_2 \leq 100$. If no random bias value is needed, then it should be negated by setting $r = 0$ or, Ec. (10) should be used instead.

With the help of the Excel "text to columns" function, for the creation of the DNN custom equation system 1 databases, the Eq. (11) was employed to generate each of the samples contained in the following .csv (comma delimited) files:

- randDnnCustomEquationSystem1/1systems_10samplesPerSys.csv
- randDnnCustomEquationSystem1/10systems_10samplesPerSys.csv
- randDnnCustomEquationSystem1/10systems_100samplesPerSys.csv
- randDnnCustomEquationSystem1/100systems_10samplesPerSys.csv
- randDnnCustomEquationSystem1/100systems_100samplesPerSys.csv

And for the ones made from the Eq. (10):

- dnnCustomEquationSystem1/1systems_10samplesPerSys.csv
- dnnCustomEquationSystem1/10systems_10samplesPerSys.csv
- dnnCustomEquationSystem1/10systems_100samplesPerSys.csv
- dnnCustomEquationSystem1/100systems_10samplesPerSys.csv
- dnnCustomEquationSystem1/100systems_100samplesPerSys.csv

For all these files, note that they try to mimic how a real database would normally be organized by a professional and in which you will encounter four columns, whose headers and purpose are the following:

1. **id:** Represents the unique identifier for the current row of the database.
2. **system_id:** Represents the unique identifier for the current system sampled. This is because the databases will contemplate having several samples for several systems that manifest the same phenomenon.
3. **dependent_variable:** Represents the output value of the current sample.
4. **independent_variable_1:** Represents the input value 1 that generated the current sample.
5. **independent_variable_2:** Represents the input value 2 that generated the current sample.

Moreover, the samples generated aimed to attempt mimicking how several real life systems behave in real life thanks to the use of the bias component r. On the other hand, each listed database was generated through a separated file which was developed in Python programming language (v3.7.1) in order to display a friendly and simple code:

- 1systems_10samplesPerSys.py
- 10systems_10samplesPerSys.py
- 10systems_100samplesPerSys.py
- 100systems_10samplesPerSys.py
- 100systems_100samplesPerSys.py

Finally, there is an additional file that has nothing to do with the generated databases, which is a MATLAB file (with .m extension), located in the directory address “dnnCustomEquationSystem1/dnn_sketch.m”, in which the code used to plot the 3D chart of Figure 1-1 was made.

Created in: June 24, 2022.

Last update in: N/A.