Function Neural Network Application Manual

MSc. Eber Jair Flores Andonegui.

eberjair@gmail.com

1. Application Overview

The application is a simple neural network demo. The neural network is designed to learn to imitate any two variables function that is continuous in a determined domain. For the details of the neural network design see the Neural Network Design document (NN Design.pdf).

This application also includes some viewers to visualize de learning process. In the next section these viewers, the options and parameters sections of the application are described.

2. Application Sections

2.1 Neural Network Viewer

This viewer shows the structure and weights between neurons, represented by colors, from -1 to 1. The neurons are arranged from bottom to top, starting with X1 and X2 for the input layer; bias unit, $S_1^{(1)}$, $S_2^{(1)}$, $S_3^{(1)}$, etc. for the first hidden layer, and so on for the next layers. For more details see the Neural Network Design document (NN Design.pdf).

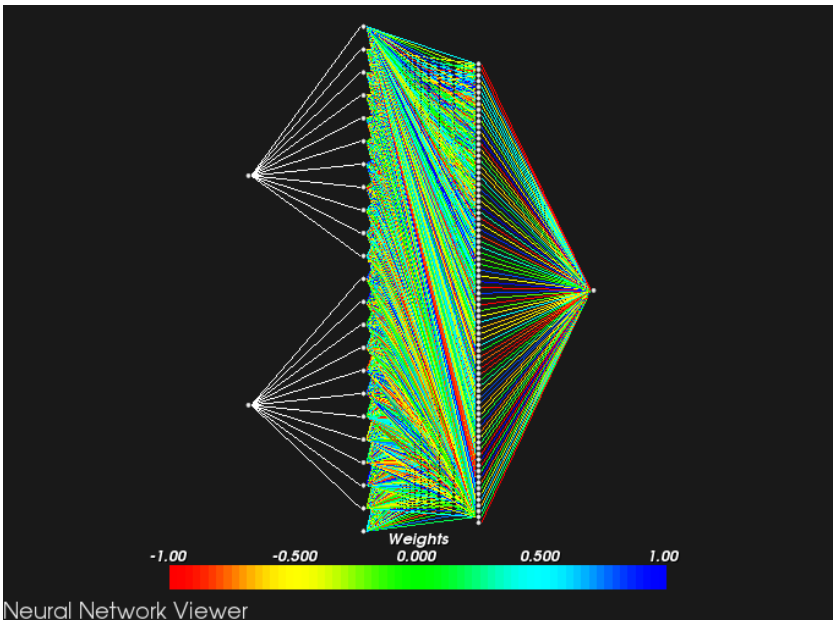

Fig 1. Neural Network Viewer

2.1.1 Neural Network Viewer Controls.

- Left click + mouse movement: Truck/Boom Camera
- Mouse wheel click + mouse movement: Truck/Boom Camera
- Right click + mouse up/down: Zoom in/out
- Mouse wheel: Zoom in/out

2.2 Error bar chart

The Error bar chart shows the last 100 mean normalized errors (neural network output error) for the last test or learning iterations or all of errors in case there are 100 iterations or less. When using the Intervals option in the Learning options, the Error bar chart will refresh every interval execution.

Fig 2. Error bar chart (initial state)

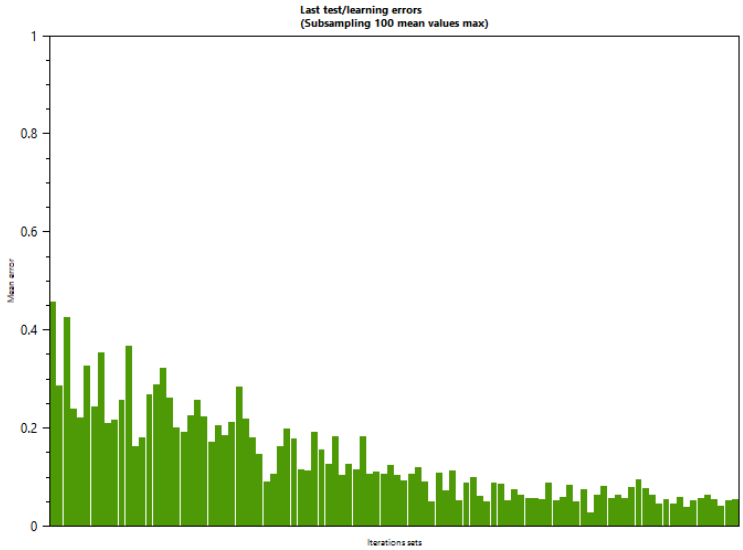

Fig 3. Error bar chart (after 1000 learning iterations)

2.3 Function to Learn Viewer and Neural Network Calculations Viewer

Function to Learn and Neural Calculations viewers are used as comparative viewers. The first one simply shows the graphic of the two variables function that the neural network should to imitate. The second one shows the graphic of the neural network calculations. Both graphics use X1 and X2 Domain defined in the Learning options as well as the X1 and X2 Intervals defined in the Visualization options to determine the domain in the X1 and X2 axis and the refinement of the visualization.

Fig 4. Function to Learn and Neural Network Calculations Viewers

2.3.1 Function to Learn and Neural Network Calculations Viewers Controls.

- Left click + mouse movement: Pan/Tilt camera
- Right click + mouse up/down: Zoom in/out
- Mouse wheel: Zoom in/out
- Mouse wheel click + mouse movement: Truck/Boom Camera
- R Key: Reset camera
- W Key: Wired visualization (edges)
- S Key: Surface visualization (faces)

2.4 Weights

The section Weights shows the current weights of the Neural Network (from -1 to 1) and allows to change them. See the Neural Network Design document (NN Design.pdf) for the details about what weight corresponds to which connection.

2.4.1 Options

- Save: Saves current weights.
- Load: Loads file with previously saved weights.
- Randomize: Generates random weights.

Fig 5. Weights options

2.5 Visualization options

The visualization options are specific for the Function to Learn and Neural Network Calculations viewers.

Sync visualization indicates whether or not to synchronize the camera for both viewers.

X1 and X2 Intervals define the refinement of the graphic.

Refresh, as its name indicates, refreshes both graphics using the current values of X1 and X2 Intervals as well as the X1 and X2 domains defined in the Learning options.

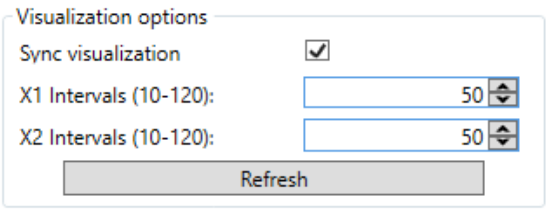

Fig 6. Visualization options

2.6 Learning options

2.6.1 The Learning process

The Neural Network uses supervised learning method with backpropagation. Random values within the domain are generated and then are normalized; the neural network calculates the normalized output using the current weights and then calculates the absolute error that is used, along with more errors, for the Error bar chart.

The backpropagation process is executed once per each input-output set. It adjusts the current weights using max descend optimization to minimize an error function. This

optimization method uses the gradient of the error function and a factor that multiplies the gradient, the gradient factor.

See the Neural Network Design document (NN Design.pdf) for the details of the Learning Process.

All this process is repeated as necessary. Each time this process is executed is known as Iteration.

2.6.2 The options

- Function: Select the function to imitate.
  To add more functions you do it in the method AddFunctions in MainWindow.xaml.cs along with its implementation in Functions/FunctionsImplementatios.cs
- X1 and X2 Domain: Min and max values for the input values.
- Y Range: Min and max values for the function range within the domain previously defined. This is necessary since the output of the neural network is supposed to be normalized.
- Iterations: Times the backpropagation process is executed.
- Grad factor type: Three options are available; $\eta = \frac{0.1}{t}$, $\eta = \frac{0.1}{\ln(t+1)}$, where t is the iteration in the process, or a Constant value defined in the Grad factor option.
- Grad factor: Value for the gradient factor in case that the Gradient factor type selected is Constant.
- Interval visualization: Frequency (number of iterations) on which the Neural Network Calculations Viewer is refreshed during the learning process execution.
- Execute: Executes the learning process with the specified options.

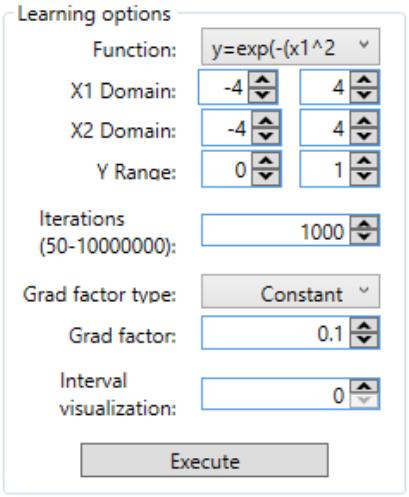

Fig 7. Learning options

2.7 Test options

This section is simply intended to test multiple random-generated input values using the weights defined for the neural network at any time. The errors in the test execution are shown in the Error bar chart.

The Number of executions is the number of input-output sets generated.

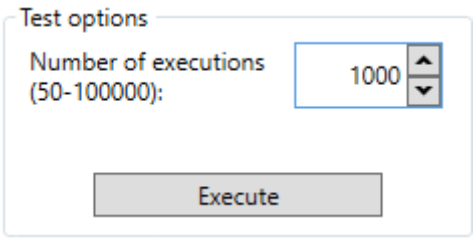

Fig 8. Test options